Object Detection Based on Plane Segmentation and Features Matching for a Service Robot

António J. R. Neves, Rui Garcia, Paulo Dias, Alina Trifan

Abstract—With the aging of the world population and the continuous growth in technology, service robots are more and more explored nowadays as alternatives to healthcare givers or personal assistants for the elderly or disabled people. Any service robot should be capable of interacting with the human companion, receive commands, navigate through the environment, either known or unknown, and recognize objects. This paper proposes an approach for object recognition based on the use of depth information and color images for a service robot. We present a study on two of the most used methods for object detection, where 3D data is used to detect the position of objects to classify that are found on horizontal surfaces. Since most of the objects of interest accessible for service robots are on these surfaces, the proposed 3D segmentation reduces the processing time and simplifies the scene for object recognition. The first approach for object recognition is based on color histograms, while the second is based on the use of the SIFT and SURF feature descriptors. We present comparative experimental results obtained with a real service robot.

Keywords—Service Robot, Object Recognition, 3D Sensors, Plane Segmentation.

I. Introduction

SERVICE robots are robots designed to assist humans on daily tasks in domestic environments. A basic service robot has to be capable of avoiding obstacles while navigating in known and unknown environments, recognizing and manipulating objects and understanding commands from humans. A large number of this type of robots have been developed over the last decades by academies and research groups. The results obtained through the conducted experiences will undoubtedly shape the robots of tomorrow in fields such as Face Recognition, Speech Recognition, Sensor Fusion, Navigation, Manipulation, Artificial Intelligence and Human-Robot Interaction.

To be able to interact with the environment, the robots can use a variety of sensors. One of the most rich types of sensors is a digital camera. Computer vision systems provide the robots the ability to understand the environment around them, to detect objects and to classify them. Performing this task in a time-constrained manner requires an efficient vision system and the processing time is a relevant issue to be considered when such vision system is implemented.

The detection of objects is of paramount importance in service robots, especially in manipulation tasks [1]. The main differences from 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.

This paper studies two approaches for object detection for a service robot based on color histograms and feature detection algorithms. These approaches have been tested on a robotic platform designed as a service robot and the processing times achieved, as well as the accuracy of the object detection prove that these solutions are suitable for real-time operation of such a robot.

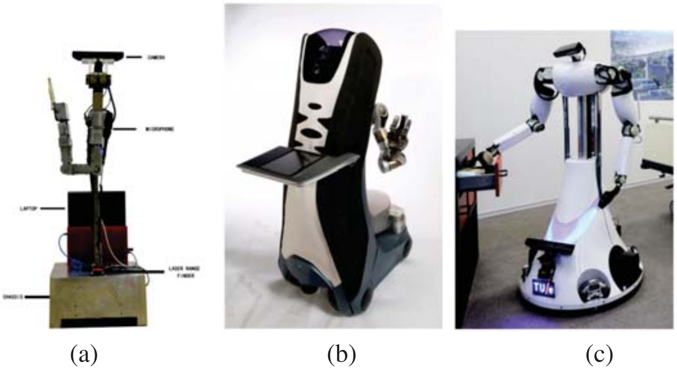
This paper is structured in six sections, first of them being this Introduction. Section II focuses on related work done in this field. In Section III we introduce the robotic platform on which the vision system has been implemented and tested. Section IV describes the algorithms that we studied and the proposed approach and in Section V we present experimental results. Finally, Section VI concludes the paper, followed by the acknowledgement of the institutions that supported this work.

II. Related Work

RoboCup Federation [2] is an international initiative devoted to promote research in the areas of robotics through robotic competitions that take place annually. Research areas as Artificial Intelligence, Computer Vision, Machine Learning, Multi-agent Systems and Biped Walking, just to name a few, have grown tremendously in the last years due to these competitions. One of the RoboCup Leagues, the @Home League (Fig. 1), is dedicated to service robots, that have to navigate through a household environment (Fig. 2) and execute some of the most diverse tasks, such as interacting with a person, following the same person through a crowded room or fetching objects that the person has indicated through oral messages. This league, through its competitions, provide great scenarios for testing and improving current developments in the area of domestic and service robotics. The work presented in this paper is intended as a contribution for a service robot that participates in the RoboCup@Home competitions and that has been developed at the University of Aveiro.

Some of most recent approaches used in service and domestic robots 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 objects' positions, and then it computes different 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 [3] and SURF [4]) and/or the use of 3D feature descriptors such as feature histograms obtained from range images (e.g., PFH [5] and VFH [6]). In most of the cases, the robotic platforms

António J. R. Neves, Rui Garcia, Paulo Dias, and Alina Trifan are with the Universidade de Aveiro, IEETA/DETI - IRIS Laboratory, Aveiro, Portugal (e-mail: {an, ruigarcia, paulo.dias, alina.trifan}@ua.pt).



(a) (b) (c)

Fig. 1 Different robots that participate in the RoboCup@Home League: (a) KeJia (b) Care-O-bot (c) Amigo

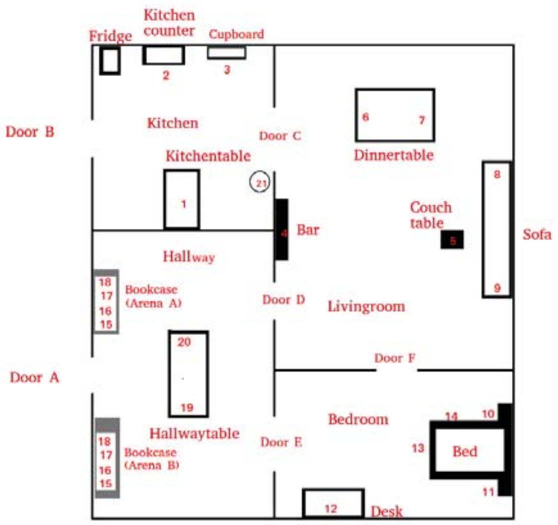


Fig. 2 A typical map of the household in which service robots competing in RoboCup@Home have to perform

are equipped with RGB and/or RGB-D cameras for data acquisition.

In [7], an approach for the detection and localization of table-top objects including bottles, cups, etc. is presented. The depth image acquired by a 3D sensor is first transformed and segmented, then the largest horizontal plane is extracted using Point Cloud Library (PCL) [8] and point clouds above it are clustered into different pieces. Next, SURF feature matching against the stored features are applied to each piece. The one with highest match above certain threshold is considered a recognition. At last, to further enhance the detection performance and decrease false positive rate, each recognized cluster is checked and the ones that vary too much in size are filtered out.

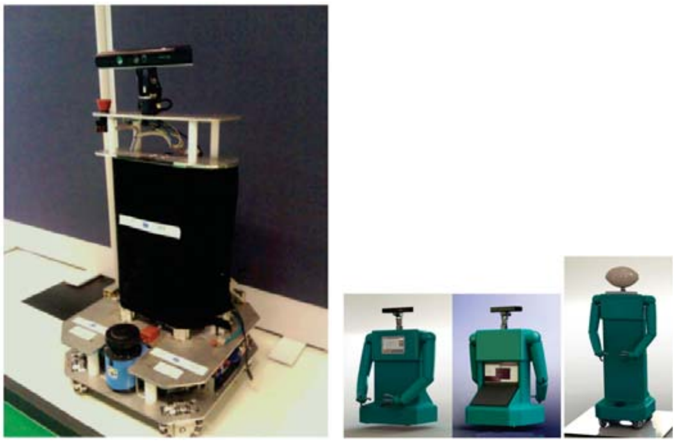
A surface reconstruction method based on Growing Neural Gas (GNG) [9] in combination with a shape distribution-based descriptor is proposed in [10] to reflect shape characteristics of object candidates. Beneficial properties provided by the GNG such as smoothing and denoising effects support a stable description of the object candidates, which also leads towards a more stable learning of categories. Based on the presented descriptor a dictionary approach combined with a supervised shape learner is presented to learn prediction models of shape categories. A classification accuracy of about 90% and a sequential execution time of less than two seconds for the categorization of an unknown object is achieved.

The vision system presented in [11] has a pipeline made of a sequence of different algorithms. Under normal operation, the first component is an object segmentation module. Objects on top of a horizontal plane are segmented using the point cloud library - PCL. This module also approximates the size of the segmented objects and based on the size selects a subset of candidate objects from the object database. The segmented objects and the related color image regions are given to the next module which is based on a custom implementation of the linemod algorithm [12]. This algorithm recognizes objects based on 2D and 3D gradient templates and color information. The module refines the subset of candidate objects provided by the segmentation module. The last module is based on the objects of daily use finder perception system, which is implemented by the ROS ODU finder package [13]. This system aims at recognizing textures objects by adopting SIFT features using vocabulary trees. The object detection and recognition system has a probabilistic nature. Each of the candidate objects given to the next module is associated with a probability. The module refines the probabilities and removes unlikely object candidates. The last module in the pipeline provides the object position together with a probability mass function over the possible class labels to the world model. Another approach capable of running in real time is the one based on Multi-resolution Surfel Maps presented in [14]. These maps can be used to model the environment and localization or to obtain 3D object models from multiple views and track these in the camera images. In addition to the recognition of known object instances, which is based on SURF features and color histograms, methods for 3D semantic mapping have been developed based on RGB-D SLAM and random forest object-class segmentation [15].

III. Hardware Platform

The CAMBADA@Home [16] is the RoboCup@Home team of University of Aveiro, Portugal. The project was created in January 2011 following the team past experience in robotic soccer. The CAMBADA@Home platform (Fig. 3) is designed as a three layer mechanical/electronical platform which can accommodate in an effective way the number of sensors and actuators needed to participate in the RoboCup@Home challenges. The vision system is located on top of the robot and uses a Kinect sensor [17].

The CAMBADA@Home robot is equipped with a Laser Range Finder used for mapping and obstacle detection. To perceive the world that surrounds the robot a RGB-D camera (Microsoft Kinect camera) is used to capture color and depth images, for the detection of people and objects.



(a) (b)

Fig. 3 (a) The current CAMBADA@Home platform (b) The final platform design

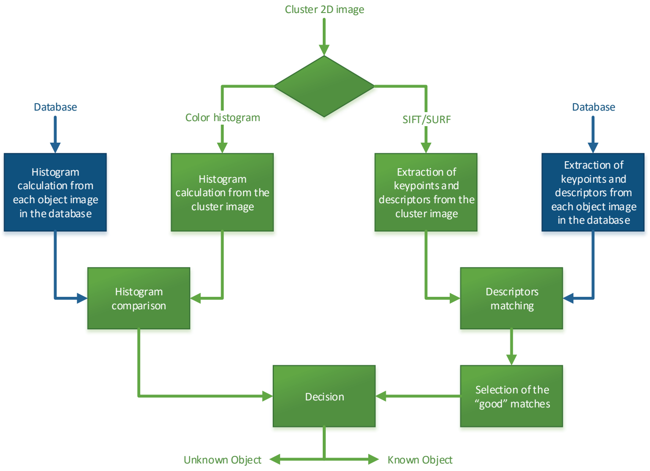

Cluster 2D image
Database
Color histogram
SIFT/SURF
Database
Histogram calculation from each object image in the database
Histogram calculation from the cluster image
Extraction of keypoints and descriptors from the cluster image
Extraction of keypoints and descriptors from each object image in the database
Histogram comparison
Descriptors matching
Decision
Selection of the "good" matches
Unknown Object
Known Object

Fig. 5 Proposed software architecture

The algorithms that will be presented in the following section have been implemented using the C++ Programming Language and they run on the processing unit of the robot, which is an Intel Core i5-3340M CPU @ 2.70GHz 4 processor, running Linux (distribution Ubuntu 14.04. LTS Trusty Tahr) . As all the previous work on the CAMBADA@Home robot is based on the ROS [13] framework, this vision system has to be compatible with it. For the processing of the point clouds, the PCL library has been used [18].

Before the development of this system, an heuristic that only objects that are placed on a plane, such as on a table or on the ground, should be considered has been defined. Therefore, the developed vision system should be able to identify and segment horizontal planes. With this assumption, two physical conditions are imposed to the robot: the Kinect must be slightly inclined towards the ground direction and it should be positioned above the height of the plane, in order to guarantee that the plane can be correctly seen (Fig. 4).

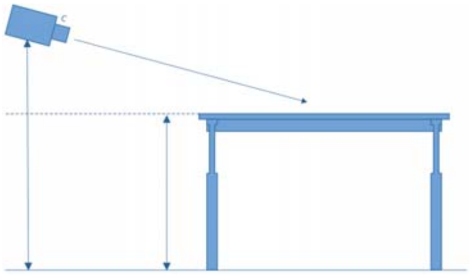


Fig. 4 Kinect position and orientation relative to a plane

IV. Proposed Approach

The workflow of the proposed approach is presented in Fig. 5

The developed vision system is provided with the three visual description algorithms that will be described next (color histograms, SIFT and SURF). Only one algorithm can be used at a time and it can be chosen at the beginning of the system execution. In the above image, the components in blue are related to processes that run only once at the beginning of the system execution. All the other components/processes are executed for each found object and will be detailed in the following subsections.

The system subscribes a topic from the ROS Kinect driver and in the first message received from the topic, i.e. in the first iteration, the system performs the calculation of the rotation vectors that will be used in the pre-processing of the point cloud provided. The first step is the pre-processing of the received point cloud. After that, the result of this process is passed to the segmentation step. If the segmentation is correctly performed, a process of extraction of the objects lying on the plane starts. In the end, a set of objects is given to the next part of the developed system.

A. Image Segmentation

Segmenting the acquired image based on horizontal plans is the first step in the pipeline implementation of the vision system that we propose. By doing so, the processing time is considerably reduced, and so is the number of possible false positives in what concerns object recognition. Fig. 6 shows the workflow performed by the system in order to segment a plane and extract the objects placed on it.

The point cloud provided by the Kinect ROS driver comes in the form of an organized 2D image of 640 rows by 480 columns, resulting in a total of 307200 three dimensional points. Two filtering processes with the objective of reducing the number of points and consequently the processing speed, resulting in a larger efficiency of the system have been implemented. The first filtering process applied to the input data is a voxel grid filter, provided by the PCL library. The purpose of this filter is the downsampling of the given point cloud, maintaining all the scene geometry but reducing the cloud size, i.e., the number of points. This filter creates a 3D voxel grid, that is a set of small 3D boxes in space over the input point cloud data. Then, in each voxel (i.e., 3D box), all the points will be approximated/downsampled with their centroid.

After the application of the voxel grid filter, a second filtering process is applied. The objective of this filter is to

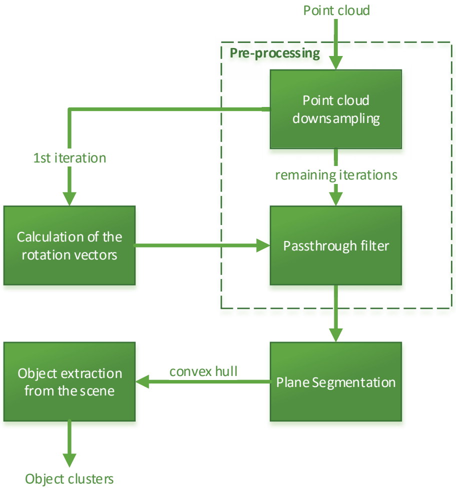


Fig. 6 Workflow for plane segmentation and object extraction

remove all the points outside the defined height limits. We iterate through the entire input point cloud once, filtering and removing all the points outside the specified interval. In this case, the threshold is defined by the user before the run of the application and represents the maximum height, in Y-axis, that is considered. As this filter acts through a defined axis and the input point cloud is not aligned with the XZ plane, the cloud must be oriented and aligned using the rotation vectors calculated in the first iteration. To align the cloud, four transformations are executed: three rotations (one per axis) and one translation in the Y-axis. The purpose of the last transformation is to translate the cloud, so that the minimum point on the Y-axis coincides in Y = 0. In the end, the pre-processed point cloud is aligned again to its original position.

After the pre-processing of the provided point cloud, the next step is the segmentation of the dominant plane presented in the scene. This is done using the iterative method known as RANSAC (RANdom SAmple Consensus) [19]. A set of indices are given as the segmentation results. These indices are then used to extract the segmented plane from the given point cloud. All the points belonging to a given cloud are stored in an auxiliary point cloud. After the extraction, a convex hull from the plane is calculated. A convex hull of a given set of points in the plane is the smallest convex polygon that contains all the points of it (Fig. 7).

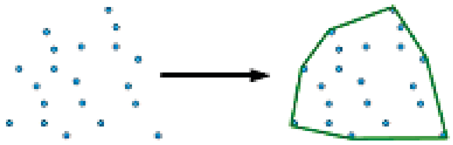


Fig. 7 An example of a convex hull of a set of points

Fig. 8 (a) shows, in red, an example of a plane segmentation. Fig. 8 (b) presents an extracted plane resulted of a segmentation.

After the process of segmentation, the result is an auxiliary point cloud with all the points inherent to the segmented plane and the corresponding convex hull. We consider that all the points lying on the segmented plane belong to graspable objects. With this information, the next step is to filter the point cloud resulting from the application of voxel grid filter, in order to get only the points that satisfy the previous consideration. A clustering algorithm has to be applied to identify individual objects. The clustering is essential to subdivide the point cloud in subsets of points (clusters) that possibly map an object. In order to reduce the overall processing time, we improved a method available in the PCL library. This method needs to divide the given point cloud into smaller parts before the clustering. A simple approach can be implemented by making use of a 3D grid subdivision. For this, a Kd-tree representation of the input point cloud has to be created.

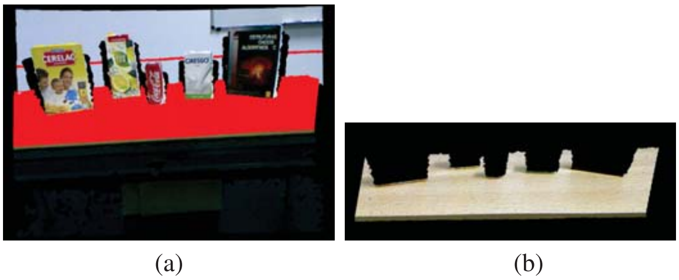


Fig. 8 (a) An example of a plane segmentation (b) the resulting image

So far, all the processing was performed in the 3D space. As stated in the beginning, the developed vision system will only explore 2D classification algorithms and all the objects will be treated as 2D objects. Because of this, each of the found clusters (potential objects) must be projected to the 2D space.

Fig. 9 illustrates the workflow performed by the system in order to project all the clusters to the 2D space.

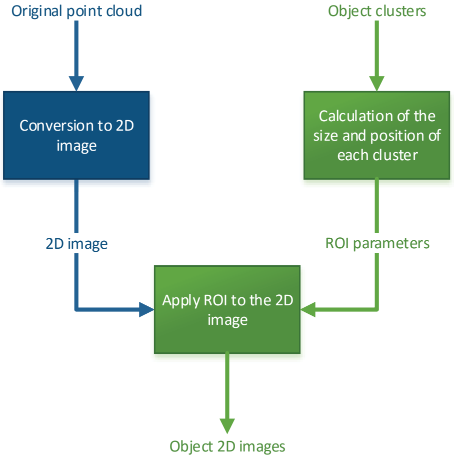


Fig. 9 Workflow for projecting the 3D clusters to the 2D space

A simple solution to form a 2D image of a cluster is to iteratively project all the 3D points to the 2D space. This could be the solution applied if the point cloud of the cluster was an organized point cloud. But, with the application of all the steps described previously, the resulting point clouds became unorganized. This is an implication of using the

Open Science Index, Computer and Information Engineering Vol:10, No:4, 2016 publications.waset.org/10004286.pdf

filters applied in the pre-processing process and the use of the clustering/extraction algorithm. The Kinect ROS driver publishes a topic that provides an organized point cloud with RGB information. This point cloud is registered with color information by the driver, which means that a direct correspondence from the 3D space to 2D space can be easily established. Then, the chosen approach was the calculation of a region of interest (ROI), in the 2D image of the original point cloud, that fits all the points of a cluster. So, two things are needed: the 2D projection of the original point cloud and the parameters to define a ROI. The first step is the cloud conversion/projection to a 2D image.

The second step is the calculation of the size and position of the cluster in the original point cloud, to find the necessary parameters to define a ROI. As the point cloud of each cluster is unorganized and subsampled, a direct correspondence can not be established. The chosen approach was to iteratively project all the 3D points of the cluster to pixels, in order to find the maximum and minimum values in the 2D space, i.e., the maximum and minimum values of the X and Y coordinates. The final step is to apply the ROI on the 2D image of the original cloud. This step acts as a filter: it creates a sub image, with the size of the ROI, where the pixels that fit outside the ROI are discarded, resulting in a 2D projection of a specific cluster.

Up to this point, the potential objects present in the captured scene were detected and projected to the 2D space. The developed vision system has to be capable to recognize these objects. In order to satisfy this requirement, a set of images has to be passed as input to the system (this set of images is referred hereby as an image database). The database contains the list of object images that the system has to be capable to recognize, i.e., the objects that the system knows.

B. Color Histograms

In image processing, a color histogram is a representation of the distribution of colors in an image. For digital images, a color histogram is a simple histogram that shows the color level for each individual color channel. If the set of possible color values is sufficiently small, each of those colors may be placed on a range by itself; then the histogram is merely the count of pixels that have each possible color. Most often, the space is divided into an appropriate number of ranges, often arranged as a regular grid, each containing many similar color values. Like other kinds of histograms, the color histogram is a statistic that can be viewed as an approximation of an underlying continuous distribution of colors values. The histogram provides a compact summarization of the distribution of data in an image.

Using color histogram as a stable representation over change in view has been widely used for object recognition. The OpenCV library provides a method [20], first introduced by Swain and Ballard [21] and further generalized by Schiele and Crowley [22], that provides the ability to compare two histograms in terms of some specific criteria for similarity. The vision system that we propose iterates once through the database and constructs the color histograms of all images. When a new frame arrives, its color histogram is calculated and compared against the ones of the images of the database. Four comparison methods have been used and experimental results are presented in the next Section of this paper for each of these methods:

- Correlation (also denoted as M1 hereby):

$$d_c(H_1,H_2) = \frac{\sum_i {H_1}'(i)\cdot {H_2}'(i)}{\sqrt{\sum_i {H_1}'^2(i)\cdot {H_2}'^2(i)}} \tag{1}$$

where $H'_k(i) = H_k(i) - (1/N)(\sum_j H_k(j))$ and N equals the number of bins in the histogram.

For correlation, a high score represents a better match than a low score. A perfect match is 1 and a maximal mismatch is –1; a value of 0 indicates no correlation (random association).

- Chi-square (also denoted as M2 hereby):

$$d_{cs}(H_1,H_2) = \sum_i \frac{(H_1(i) - H_2(i))^2}{H_1(i) + H_2(i)} \tag{2}$$

For Chi-square, a low score represents a better match than a high score. A perfect match is 0 and a total mismatch is unbounded (depending on the size of the histogram).

- Intersection (also denoted as M3 hereby):

$$d_i(H_1,H_2) = \sum_i min(H_1(i), H_2(i)) \tag{3}$$

For histogram intersection, high scores indicate good matches and low scores indicate bad matches. If both histograms are normalized to 1, then a perfect match is 1 and a total mismatch is 0.

- Bhattacharyya distance (also denoted as M4 hereby):

$$d_b(H_1,H_2) = \sqrt{1 - \sum_i \frac{\sqrt{H_1(i)\cdot H_2(i)}}{\sqrt{\sum_i H_1(i)\cdot \sum_i H_2(i)}}} \tag{4}$$

For Bhattacharyya matching, low scores indicate good matches and high scores indicate bad matches. A perfect match is 0 and a total mismatch is a 1.

C. Descriptors

Scale Invariant Feature Transform (SIFT) [3] is a popular image matching algorithm in computer vision, used to detect and describe local features in an image and it is applicable in areas as object recognition, video tracking, image stitching and 3D modeling. The SIFT descriptor is invariant to rotations, translations and scaling transformations in the image domain. It is also robust to changes in illumination, noise and perspective transformations.

For any object in an image, interesting points on the object can be extracted to provide a feature description. In the SIFT algorithm, this points are extracted (termed keypoints) from a reference image and stored in an appropriate structure. Once found all keypoints, the algorithm computes a descriptor vector for each keypoint. The resulting descriptors can be compared with the others descriptors obtained from different images, in order to found matching pairs.

Speeded Up Robust Feature (SURF) [4] is a local feature detector and descriptor and it is commonly used in tasks such as object recognition, registration, classification and 3D reconstruction. It is partly inspired by the SIFT method and it is also invariant to scale, rotation and translation. The SURF algorithm uses integral images in the convolution process, useful to speed up this method. The box-space is constructed by using box filters approximation, by convolving of the initial images with box filters at several different discrete size. To select interest point candidates, the local maxima of a Hessian matrix is computed and a quadratic interpolation is used to refine the location of candidate keypoints. Contrast signs of the interest point are stored to construct the keypoint descriptor. Finally, the dominant orientation of each keypoint is estimated and the vector descriptor is computed.

For SIFT and SURF implementations, the descriptors matching component can produce a large number of matches. For this reason, a selection process of “good” matches had to be implemented, where a reference threshold is applied and only matches with a distance less than that threshold are considered a “good” match.

The vision system that we propose iterates once through the database and extracts the keypoints and descriptors from each object image in the database. The same process occurs for each new frame. The final objective if to select the best candidate match for each keypoint. It is important to select an algorithm to perform the matching as quickly as possible and in an efficient way. In this work, two descriptors matching algorithms were studied: the Brute Force matcher and the Fast Library for Approximate Nearest Neighbor (FLANN) [23] .

- The Brute-Force matcher takes the descriptor of one feature in the first set and matched it with all other features in second set using some distance calculation, returning the closest one.
- FLANN is a algorithm for performing fast approximate nearest neighbor searches in high dimensional spaces. It contains a collection of algorithms found to work best for nearest neighbor search and a system for automatically choosing the best algorithm and optimum parameters depending on the dataset.

V. Experimental Results

To evaluate the efficiency of the developed vision system for a service robot several experiences were conducted with the CAMBADA@Home robot. Different objects were selected and several acquisitions were made using the Kinect installed on the robot. Examples of objects used in these experiments are presented in Fig. 10.

Table I presents the results of the object detection approach based on color histograms. In this experimental test, a database with three categories of images were used (Fig. 10) and three scenarios were tested, each one with only one object placed on the top of a table. The gray cells represents the best correspondence between each scenario and each method.

As it can be seen in Table I, M2, M3 and M4 are difficult to use for the detection of the milk package. The correlation method however, leads to good object recognition results for each of the three scenarios.

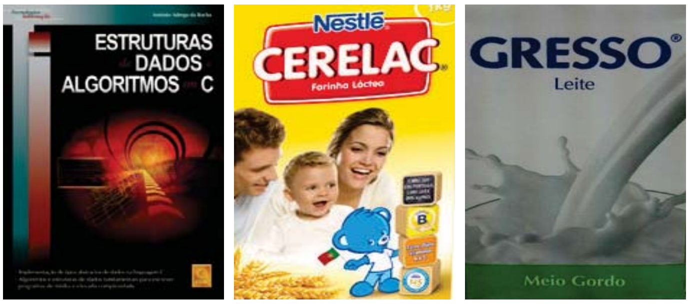


Fig. 10 Selected objects for the tests

To study the efficiency of the descriptors matching algorithms, several tests had been performed. The first was the calculation of the number of matches and “good” matches between the objects in the test scenario and the database. The scenario chosen was a table where the objects were placed and the robot performing the movement illustrated in the Fig. 11, varying the scale and rotation of the camera relatively to the position of objects.

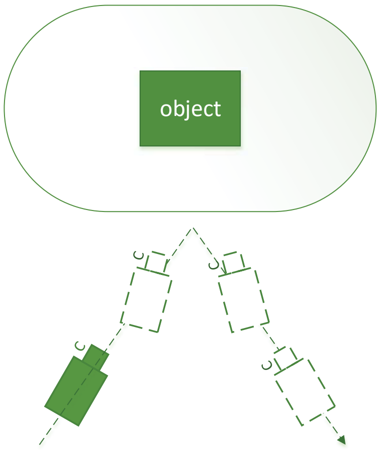


Fig. 11 The green rounded rectangle represents a table with an object. The dashed line represent the movement of robot. Two details are to be noted: (1) variation in the scale; (2) variation in rotation

Figs. 12 and 13 present SIFT results in terms of number of matches and number of “good matches” that have been obtained for the same three objects. The distribution of “good matches” follows the distribution of the original matches. It can be notted that only a small number of matches can be extracted for the milk package and this is due to the lack of texture and details of this object.

Figs. 14 and 15 present SURF results in terms of number of matches and number of “good matches”. This approach leads to even an inferior number of matches for all the objects, most visible in the case of the milk package and of the book as well. This leads to smaller recognition rates, as it will be detailed next.

Table II presents results on the efficiency of the object recognition processes previously described in terms of processing time, regarding the use of histogram comparison, SIFT and SURF.

As can be seen in these results presented the processing time

TABLE. I
COLOR HISTOGRAM COMPARISON - STUDY OF METHODS

	Method 1			Method 2			Method 3			Method 4		
Scenario	Milk	Cereals	Book	Milk	Cereals	Book	Milk	Cereals	Book	Milk	Cereals	Book
1	0.0059	0.0003	-0.0009	142430	71699	50965	1753	2026	2837	0.759	0.757	0.765
2	-0.0067	0.9492	0.0288	125112	16161	1132710	2072	12961	7480	0.837	0.314	0.582
3	-0.0038	0.086	0.6521	113634	75613	25573	1808	6897	13258	0.858	0.633	0.332

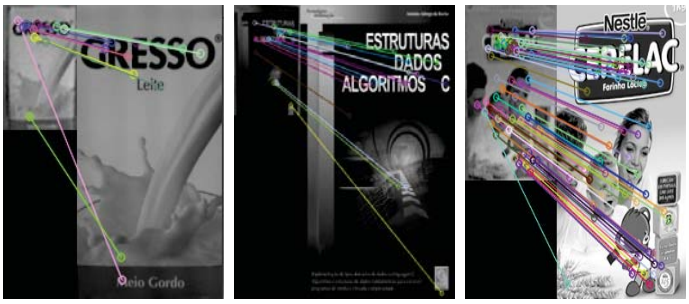
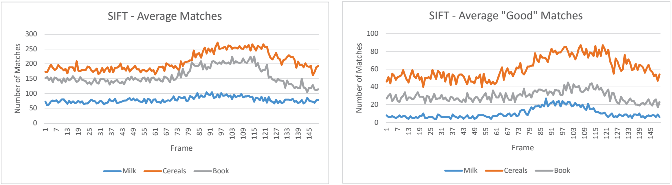


Fig. 12 SIFT descriptors matching: Colored lines represent the “good” matches



(a) (b)

Fig. 13 SIFT descriptors matching: (a) Number of matches; (b) Number of “good matches”

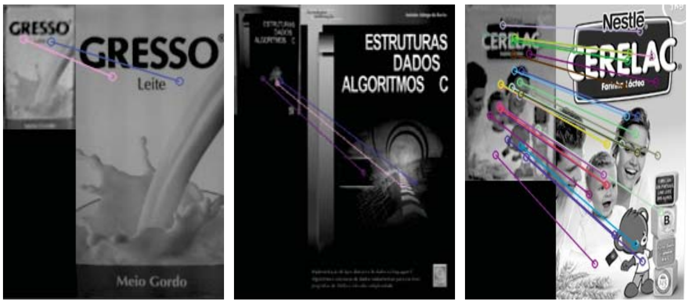
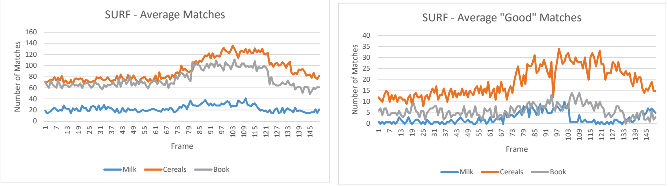


Fig. 14 SURF descriptors matching: Colored lines represent the “good” matches



(a) (b)

Fig. 15 SURF descriptors matching: (a) Number of matches; (b) Number of “good matches”

TABLE. II
EXPERIMENTAL RESULTS: EFFICIENCY IN TERMS OF PROCESSING TIME (MS) OF THE HISTOGRAM COMPARISON PROCESS, SIFT AND SURF IMPLEMENTATIONS; 150 FRAMES WERE ANALYZED. COLUMN “CLASS.” MEANS THE CLASSIFICATION TIME

	Histogram		SIFT		SURF	
Scenario	**Class.**	**Total**	**Class.**	**Total**	**Class.**	**Total**
1 object	1.428	41.269	24.428	64.437	13.698	52.142
2 objects	2.887	45.013	39.782	85.062	20.678	63.492
3 objects	5.259	48.893	57.207	102.119	30.574	74.262
4 objects	7.321	50.061	74.686	119.283	36.512	79.891
5 objects	9.913	55.925	88.623	132.823	39.137	83.138

of each classification algorithm is dependent on the number of the clusters and it is higher when there are more objects on the table. The most efficient algorithm, i.e., the one with the smaller processing time is the histogram comparison algorithm and the less efficient is the SIFT implementation. The SURF implementation leads to smaller processing times also because the detection rate of this approach is smaller than the one based on the SIFT algorithm. Several scenarios were recorded for further study of these methods:

- Scenario 1: All objects are present in the scene;
- Scenario 2: Missing Book;
- Scenario 3: Missing Cereals box;
- Scenario 4: Missing Milk package;

Table III presents efficiency results regarding processing time for each of the 4 scenarios using the SIFT approach. The most time consuming operation is the classification step, which amount to more than half of the processing time for each of the presented scenarios.

TABLE. III
EFFICIENCY OF THE DEVELOPED SYSTEM REGARDING PROCESSING TIME (MS) OF EACH BLOCK USING THE SIFT ALGORITHM; 250 FRAMES WERE ANALYZED

Scenario	Voxel	Pre-proc.	Seg.	Clust.	Projec.	Class.	Total
1	31.691	3.596	3.211	19.502	2.203	149.455	213.745
2	31.871	3.714	2.751	15.167	2.133	109.979	169.351
3	32.080	3.587	2.911	16.443	2.306	109.418	170.158
4	31.998	3.949	3.032	19.011	2.292	122.926	187.077

In terms of detection accuracy, Table IV presents results of detection rates for each of the 4 scenarios for the SIFT approach. As stated before, a larger number of matches leads to a higher recognition rate. However, the milk package, which was the object with less matches, has been recognized on more than 75% frames in each of the four scenarios. A small number of false positives can be detected in the scenarios 3 and 4.

TABLE. IV
EXPERIMENTAL RESULTS: DETECTION RATE REGARDING THE SIFT ALGORITHM, GRAY CELLS CORRESPOND TO FALSE POSITIVES. 250 FRAMES WERE ANALYZED.

Scenario	Book	Cereals	Milk
Scenario 1	91.2 %	99.2 %	75.6 %
Scenario 2	0%	100 %	95.2 %
Scenario 3	86.8 %	3.6 %	92.8 %
Scenario 4	88.4 %	98.8 %	1.2 %

Table V presents efficiency results regarding processing time for each of the 4 scenarios the SURF approach. Again, the most time consuming step is the classification process.

TABLE. V
EFFICIENCY OF THE DEVELOPED SYSTEM REGARDING PROCESSING TIME (MS) OF EACH BLOCK USING THE SURF ALGORITHM; 250 FRAMES WERE ANALYZED.

Scenario	Voxel	Pre-proc.	Seg.	Clust.	Projec.	Class.	Total
1	31.649	3.312	3.131	18.327	2.267	101.705	164.268
2	30.123	3.278	2.727	14.261	2.136	72.559	128.259
3	31.002	3.417	2.719	15.946	2.191	72.219	130.576
4	31.433	3.646	3.015	18.579	2.138	80.303	142.492

In terms of detection accuracy, Table VI presents results of detection rates for each of the 4 scenarios for the SURF approach. The recognition rate for each of the objects has slightly decreased when compared to the SIFT approach. However, the percentages of false positive is close to 0.

TABLE. VI
EXPERIMENTAL RESULTS: DETECTION RATE REGARDING THE SURF ALGORITHM, GRAY CELLS CORRESPOND TO FALSE PPOSITIVES. 250 FRAMES WERE ANALYZED.

Scenario	Book	Cereals	Milk
Scenario 1	90 %	97.6 %	72 %
Scenario 2	0%	99.6 %	83.6 %
Scenario 3	86 %	0%	74.4 %
Scenario 4	78 %	99.2 %	2%

During the course of the tests, it was observed that the best results are obtained when the robot is located in front of the objects and close to them.

As can be observed in Table VI, the SURF algorithm presents a lower processing time, when compared to the SIFT algorithm. However, in Table III and Table IV, the presented results are slightly more accurate in the SIFT algorithm than in the SURF implementation, which means that there is a compromise that a user of this system should do in order to decide whether time processing or object recognition rate is more important in a given application.

VI. CONCLUSIONS AND FUTURE WORK

We have presented in this paper two approaches for object detection for a service robot. Both approaches rely on the existence of a previously built database of images containing different instances of the objects of interest. The first method is based on the match between color histogram, while the second one is based on image descriptors. Experimental results have shown that both methods can be used in the real-time operation of a service robot. Future work will focus on the improvement of this study with more objects and under different perspectives, including the integration of 3D models.

ACKNOWLEDGMENTS

This work was developed in the Institute of Electronic and Telematic Engineering of University of Aveiro and was partially supported by FEDER through the Operational Program Competitiveness Factors - COMPETE FCOMP-01-0124-FEDER-022682 (FCT reference PEst-C/EEI/UI0127/2011) and by National Funds through FCT - Foundation for Science and Technology in a context of a PhD Grant (FCT reference SFRH/BD/85855/2012).

REFERENCES

[1] L. Martinez, P. Loncomilla, and P. Ruiz-del Solar, "Object recognition for manipulation tasks in real domestic settings: A comparative study," in *Proceedings of RoboCup 2014 Symposium*, Joao Pessoa, Brazil, July 2014.

[2] "RoboCup Federation official website," www.robocup.org, accessed: 2015-09-30.

[3] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[4] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.

[5] R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz, "Persistent Point Feature Histograms for 3D Point Clouds," in *Proceedings of the 10th International Conference on Intelligent Autonomous Systems (IAS-10), Baden-Baden, Germany*, 2008.

[6] J. Borenstein and Y. Koren, "The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278–288, 1991.

[7] R. B. Rusu, A. Holzbach, G. Bradski, and M. Beetz, "Detecting and segmenting objects for mobile manipulation," in *Proceedings of IEEE Workshop on Search in 3D and Video (S3DV), held in conjunction with the 12th IEEE International Conference on Computer Vision (ICCV)*, Kyoto, Japan, September 27 2009.

[8] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *International Conference on Robotics and Automation*, Shanghai, China, 2011 2011.

[9] Y. Holdstein and A. Fischer, "Three-dimensional surface reconstruction using meshing growing neural gas (mgng)," *The Visual Computer*, vol. 24, no. 4, pp. 295–302, 2008.

[10] R. Dwiputra, M. Füller, F. Hegger, S. Schneider, I. A. J. M. S. Loza, A. Y. Ozhigov, S. Biswas, N. V. Deshpand, A. H. I. Ivanovska, P. G. Ploeger, and G. K. Kraetzschmar, "The b-it-bots robocup@home 2014 team description paper," Joao Pessoa, Brazil, 2014.

[11] S. A. M. C. J. J. M. Lunenburg and T. T. J. Derksen, "Tech united eindhoven @home 2014 team description paper," Joao Pessoa, Brazil, 2014.

[12] T. D. Jager, "Robust object detection for service robotics," PhD thesis, Utrecht University, 2013.

[13] "Robot Operating System official website," www.ros.org, accessed: 2015-09-30.

[14] J. Stückler and S. Behnke, "Multi-resolution surfel maps for efficient dense 3d modeling and tracking," *J. Vis. Comun. Image Represent.*, vol. 25, no. 1, pp. 137–147, Jan. 2014.

[15] J. Stückler, B. Waldvogel, H. Schulz, and S. Behnke, "Dense real-time mapping of object-class semantics from rgb-d video," *Journal of Real-Time Image Processing*, 2014.

[16] "CAMBADA@HOME official website," http://robotica.ua.pt/CAMBADA@HOME/, accessed: 2015-09-30.

[17] "Microsoft Kinect official website," https://dev.windows.com/en-us/kinect, accessed: 2015-09-30.

[18] "Point Cloud Library official website," http://pointclouds.org/, accessed: 2015-09-30.

[19] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[20] D. G. R. Bradski and A. Kaehler, *Learning Opencv, 1st Edition*, 1st ed. O'Reilly Media, Inc., 2008.

[21] M. J. Swain and D. H. Ballard, "Color indexing," *Int. J. Comput. Vision*, vol. 7, no. 1, pp. 11–32, Nov. 1991.

[22] B. Schiele and J. L. Crowley, "Object recognition using multidimensional receptive field histograms," in *Proceedings of the 4th European Conference on Computer Vision-Volume I - Volume I*, ser. ECCV '96. London, UK: Springer-Verlag, 1996, pp. 610–619.

[23] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *In VISAPP International Conference on Computer Vision Theory and Applications*, 2009, pp. 331–340.