J Intell Robot Syst (2015) 80:537–553
DOI 10.1007/s10846-015-0189-z

Interactive Open-Ended Learning for 3D Object Recognition: An Approach and Experiments

S. Hamidreza Kasaei · Miguel Oliveira · Gi Hyun Lim · Luís Seabra Lopes · Ana Maria Tomé

Received: 7 July 2014 / Accepted: 9 January 2015 / Published online: 31 January 2015

Abstract 3D object detection and recognition is increasingly used for manipulation and navigation tasks in service robots. It involves segmenting the objects present in a scene, estimating a feature descriptor for the object view and, finally, recognizing the object view by comparing it to the known object categories. This paper presents an efficient approach capable of learning and recognizing object categories in an interactive and open-ended manner. In this paper, "open-ended" implies that the set of object categories to be learned is not known in advance. The training instances are extracted from on-line experiences of a robot, and thus become gradually available over time, rather than at the beginning of the learning process. This paper focuses on two state-of-the-art questions: (1) How to automatically detect, conceptualize and recognize objects in 3D scenes in an open-ended manner? (2) How to acquire and use high-level knowledge obtained from the interaction with human users, namely when they provide category labels, in order to improve the system performance? This approach starts with a pre-processing step to remove irrelevant data and prepare a suitable point cloud for the subsequent processing. Clustering is then applied to detect object candidates, and object views are described based on a 3D shape descriptor called spin-image. Finally, a nearest-neighbor classification rule is used to predict the categories of the detected objects. A leave-one-out cross validation algorithm is used to compute precision and recall, in a classical off-line evaluation setting, for different system parameters. Also, an on-line evaluation protocol is used to assess the performance of the system in an open-ended setting. Results show that the proposed system is able to interact with human users, learning new object categories continuously over time.

Keywords Open-ended learning · 3D object recognition · Spin-image descriptor · Autonomous robots

S. H. Kasaei (✉) · M. Oliveira · G. H. Lim · L. Seabra Lopes · A. M. Tomé
IEETA - Instituto de Engenharia Electrónica e Telemática de Aveiro, Universidade de Aveiro, Aviero, Portugal
e-mail: seyed.hamidreza@ua.pt

M. Oliveira
e-mail: mriem@ua.pt

G. H. Lim
e-mail: lim@ua.pt

L. Seabra Lopes
e-mail: lsl@ua.pt

A. M. Tomé
e-mail: ana@ua.pt

L. Seabra Lopes · A. M. Tomé
Departamento de Electrónica, Telecomunicações e Informática, Universidade de Aveiro, Aveiro, Portugal

1 Introduction

Following the recent release of inexpensive 3D sensing devices such as Microsoft Kinect[1] and ASUS Xtion,[2] which record RGB and depth information, 3D object detection, recognition and reconstruction have become a widespread research topic. 3D object detection and recognition have been addressed in both robotics and human-robot interaction research communities.

Although many object detection and recognition methods for both 2D and 3D data have been proposed [1], the research in this field is still active. In most of the proposed systems, training and testing are separated processes [18], and since limited sample data are used for the training, the systems are unable to adapt to dynamic environments [22]. Besides this, these systems have other limitations such as the need to undergo an exhaustive procedure of manually annotating the training data-sets, or the inability to detect / recognize new or unknown objects [1], as well as the incapacity to handle object occlusion and scene clutter [28]. Real 3D scenes generally consist of several objects present in a scene. Therefore, two important issues concerning 3D object recognition in complex scenes are clutter and occlusion. Clutter is seen when points that do not belong to the target object are included in the segmentation, which affects the recognition process. Also, because many objects are present in the scene, some of them may be occluded by others. Thus, many times, the information related to a particular object is partial [29]. To cope with both occlusion and clutter, 3D data is used to facilitate the detection of complex (i.e. free form) objects [3]. The reason for this is that 3D data contains more information about the spatial positioning of objects, which in turn eases the process of segmentation. Moreover, depth data is more robust than RGB data to the effects of illumination [18]. 3D data can therefore be employed to describe the surface of the object based on its geometric properties.

Humans learn to recognize object categories ceaselessly over time. This ability allows them to adapt to new environments, by enhancing their knowledge from the accumulation of experiences and the conceptualization of new object categories [11]. Taking this as inspiration, an object perception system should process visual information continuously, and perform recognition and learning simultaneously [20].

In this paper, an interactive 3D object learning and recognition approach [13] is presented. The approach is designed to be integrated in an open-ended learning system, which is used by an autonomous service robot working in a restaurant scenario [9, 21]. This work focuses on learning and recognizing table-top objects, which can be manipulated by the robot.

The remainder of this paper is organized as follows: next, related work is presented; a general overview of the proposed interactive object learning and recognition system is described in Section 3; the detailed methodology is then explained in Sections 4 and 5; an experimental evaluation is presented and discussed in Section 6; finally, conclusions and future work are presented in Section 7.

2 Related Work

Different approaches have been proposed to solve the limitations of object learning and recognition system over the past five decades [1]. Willow Garage started a project named Object Recognition Kitchen (ORK),[3] a 3D object recognition system built on top of the Ecto framework.[4] ORK was designed to run simultaneously several traditional object recognition techniques, so that these can be combined for example using a voting scheme. Ecto is a C++ / Python computation graph framework, which can organize the computation modules in a directed acyclic graph. This means that there are some limitations in the system configuration. In ORK, the training and recognition are not simultaneous. Romea et al. [5], described an object recognition and pose estimation system. In this case, the system is decomposed into an off-line training stage and an on-line recognition stage. In the training stage, for every object, a set of images are captured from different viewpoints. Then, SIFT features are extracted for each image and stored in a database.

[1] http://www.xbox.com/en-US/kinect

[2] http://www.asus.com/Multimedia/Xtion_PRO

[3] http://wg_perception.github.io/object_recognition_core/

[4] http://plasmodic.github.io/ecto/

During the recognition stage, SIFT features are computed for the current view and matched against the training models. The authors use a Best Bin First algorithm [2] for matching. Martinez et al. [17] proposed a fast and scalable perception system for object recognition and pose estimation. The authors employed RANSAC [8] and Levenberg Marquardt algorithms to segment objects and represent these based on SIFT descriptors. Kootstra et al. [15] also proposed an active perception system for recognizing objects that are placed in cluttered and uncontrolled environments. They used a mobile robot that explores the objects by circling around them and capturing data. They also used SIFT descriptors for learning and recognition tasks.

Johnson and Herbert [12] proposed a spin-image descriptor which represents the spatial distribution of 3D points around key points. Spin-image descriptors efficiently encode the 3D geometry of the surface patches they describe. This helps to cope with recognition problems in complex scenes. Spin-images are also known to be pose invariant. Dinh and Kropac [6] proposed multi-resolution pyramids of spin images in order to improve the discriminative power of the original spin-images. This approach also speeds up the matching process.

Some authors argued that comparing objects by their local features is computationally expensive. The key idea for fast 3D object recognition, is to use mechanisms for representing objects in a highly compact and distinctive way. One possible solution to this problem is to employ a bag-of-words technique. This has been proposed in recent years. Islam et al. [10], described an object classification approach is described, in which the object representation is based on SIFT, SURF and color histograms. All these features are compacted into a histogram of words. In this case, authors use a naive Bayes classifier in the recognition stage. Liu and Zha [16] used a bag-of-words technique for optimizing the recognition process, as well as memory usage. The authors investigated the problem of efficient partial 3D shape retrieval. First, a Monte-Carlo method used to select interest points is proposed, and then, the spin-image descriptors are used to encode the geometry around the interest points. In the recognition stage, they proposed to use a dissimilarity measure based on the asymmetric Kullback-Leibler divergence.

In most of the proposed systems described above, training and recognition are separate processes, which do not occur simultaneously. However, off-line training is not suited for open-ended scenarios, because the target categories are not known in advance. Systems limited to off-line training are unable to adapt to dynamic environments. There are some approaches which support incremental learning of object categories. Yeh and Darrell [30] developed novel methods for efficient incremental learning of SVM-based visual category classifiers, and showed that, using their framework, it is possible to adapt the classifiers incrementally. Takamuku et al. [27] proposed a learning approach for object recognition based on physical interaction. In this case, the system was capable of classifying objects into three categories, according to the joint angle data obtained while the robot shook the objects. Human-robot interaction is essentially used for the gathering of supervised experiences. In particular, it can be used for teaching object categories in situations where either the object is unknown to the system or when the system is misclassifying the object. Kirstein et al. [14] proposed a lifelong learning approach for interactive learning of multiple categories based on vector quantization and a graphical user interface. The instructor could give the names of objects using the graphical user interface. Using only 2D images, they showed the system could successfully learn 5 color categories and 10 shape categories observed in 56 objects. Seabra Lopes and Chauhan [25, 26] approached the problem of object experience gathering and category learning with a focus on open-ended learning and human-robot interaction. In their approach, learning is based on multiple representations and classifiers, as well as combinations of classifiers. A meta-learning component monitors the performance of the classifiers, and reconfigures the combination of classifiers in order to achieve the best recognition performance. In addition to this, their framework allowed the user to provide the names of objects through pointing and verbal teaching actions. The user could also ask questions about the categories of objects under shared attention and, if appropriate, provide corrective feedback for them. They showed a system that starts with an empty vocabulary and can incrementally acquire object categories through the interaction with a human user. The authors also proposed a teaching protocol for evaluating the

performance of open-ended object learning and recognition approaches.

3 Overall System Architecture

In this paper, the problem of interactive open-ended object learning is addressed, which enables a robot to acquire information about unknown objects, and store the information in a perceptual memory. From a global perspective, the system is composed of several software modules such as object detector, feature extractor, object conceptualizer, user interface and object recognizer. The overall system architecture is depicted in Fig. 1. The processing cycle is triggered when the robot captures an image of the scene. The first step is object detection, which involves distance filtering, down-sampling, and clustering of object points. The object detector periodically requests from another software module called tabletop segmenter a list of all the objects currently on top of the table. The object detector module creates a new perception pipeline for every newly detected object. Each pipeline includes the object tracker, the feature extractor and the object recognizer modules. The object tracker works based on a particle filter [24] which uses geometric information as well as color and surface normal data to predict the next probable pose of the object. The object tracker sends out the object point cloud to the feature extraction module. This object point cloud is used by the feature extractor module to compute features for the given object view using 3D shape descriptors. The features of objects are kept in memory, and the user can provide category labels for those objects. Object labeling, handled via the user interface module, triggers the object conceptualizer module. In such situation, the object conceptualizer reads the current object categories from memory, as well as a set of features describing the labeled object, and creates or improves an object category. During recognition situations, a nearest-neighbor classification rule is used to estimate the category labels of the detected objects. In the following sections, each module is explained in detail.

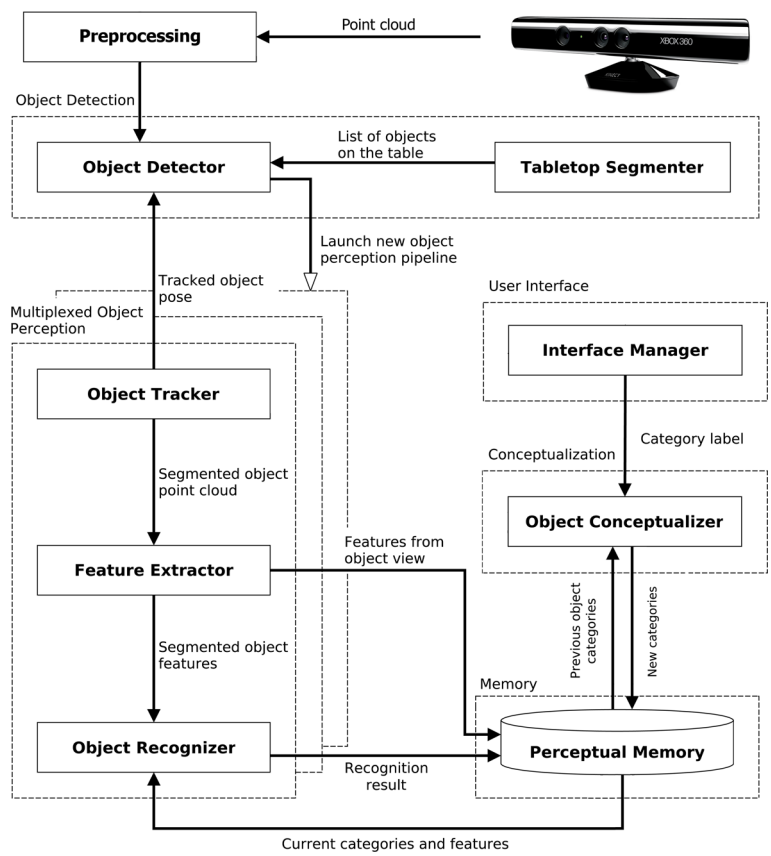

Fig. 1 Overall architecture of the proposed system

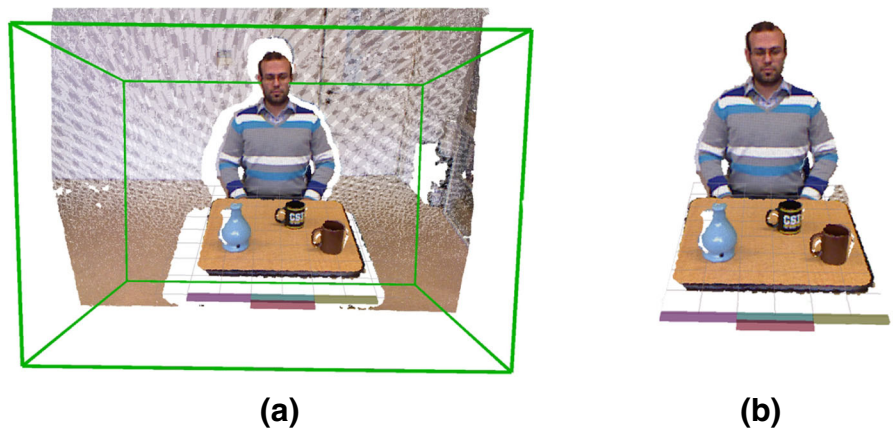

Fig. 2 An example of the pre-processing functions: **a** graphical perspective projection of the original point cloud; **b** result of the first preprocessing step

4 Shape Feature Extraction

In this section, we discuss the process of extracting 3D shape features, which are used both for conceptualization as well as for recognition. In particular, we detail the pre-processing, object detection and feature extraction components from Fig. 1. Note that, in this paper, we describe in detail the conceptualizer and recognition components of the system. For additional details on other system components see [19].

4.1 Pre-Processing

Processing massive point clouds is one of the main challenges of 3D perception systems. In dense 3D point cloud data, considering all points is computationally too expensive, and real time processing is not possible. The key idea for fast processing of massive point clouds is to use mechanisms for removing unnecessary or irrelevant data. To accomplish this, we use two separate filters that discard vast quantities of unnecessary 3D points. The first filter defines a cubic volume in 3D, which defines the region of interest in which we consider relevant points should be. In our current setup, we use a table which is approximately one meter away from the camera / robot. Using this information, we define the size of the cubic volume to include a typical table in front of the robot. Figure 2 shows an example of this process. In Fig. 2a, the complete point cloud is shown, along with the cube. In Fig. 2b, the filtered point cloud is displayed. This filter enables a significant reduction of the number of points.

The second filter in the pre-processing reduces the spatial resolution of points, since our approach does not require the full resolution of the sensor. To do this, the point cloud is down sampled using a voxelized grid approach.[5] The advantage of this, apart from the fact that the number of points is further reduced, is that the spatial distribution of 3D points becomes uniform.

4.2 Object Detector

After pre-processing, the next step is to find objects in the scene. Our approach assumes that objects should be placed on top of the table in order to be detected. The table is detected by finding the dominant plane in the point cloud. This is done using a RANSAC [8] algorithm. The algorithm starts by generating plane hypotheses based on three unique non-collinear points. Afterwards, distances from all points in the point cloud to the plane are computed. The plane hypotheses are then scored based on counting the number of points whose distance to the plane falls below a user-specified threshold, dt. The RANSAC algorithm is repeated for a certain number of iterations, N. In the current implementation, $dt = 20mm$ and $N = 200$. An example of the proposed table detector algorithm is illustrated in Fig. 3b. With the table detected, it is now possible to extract the points which lie directly above it. The mechanism we use to do this is called extraction of polygonal prisms.[6] After this, we have a point cloud where all the objects

[5] http://pointclouds.org/documentation/tutorials/voxel_grid.php

[6] http://docs.pointclouds.org/1.0.0/classpcl_1_1_extract_polygonal_prism_data.html

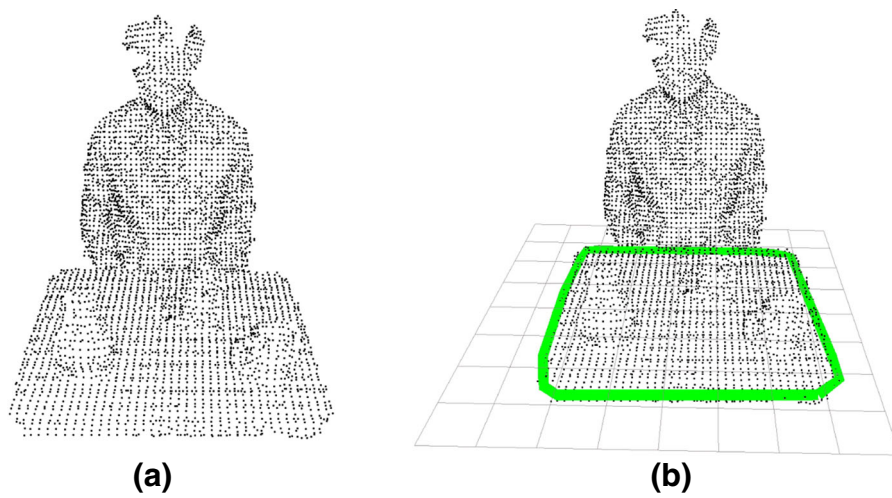

Fig. 3 **a** An example of the down sampling using a voxel grid filter; **b** result of the table detection

that are on top of the table are included. To separate each object into an individual point cloud, we use an Euclidean cluster extraction algorithm.[7] Each group of points will be treated as an object candidate. An example of the object detector module is shown in Fig. 4. In Fig. 4a, the complete point cloud is shown. Three objects are on top of the table. In Fig. f 4b, the segmented object point clouds are shown. Note that the point clouds of each object have different colors, meaning that they have been correctly segmented.

4.3 Feature Extractor

We adopt an approach to object recognition in which objects are described by local shape features called spin-images [12]. The reason why we use spin images rather than other 3D feature descriptors such as viewpoint feature histogram [23] is that we are interested in local features, rather than global. These features are pose invariant, and therefore suitable for 3D perception in autonomous robots. The feature extractor module displayed in Fig. 1 consists of two main phases: extraction of keypoints and the computation of spin images. For efficiency reasons, the number of keypoints in an object should be much smaller than the total number of points. To do that, we use a voxelized grid to subsample the object point cloud by taking only the nearest neighbor point for each voxel center (see [7] for an analysis of 3D descriptors). Figure 5a shows an example of the keypoint extraction process.

In the second stage, spin-image descriptors are computed for each keypoint in order to describe the shape surrounding the keypoint. A spin-image is a local shape histogram obtained by projecting the 3D surface points onto the tangent plane of the keypoint. The normal vector of the tangent plane is called surface normal. Then, each point is represented by a pair (α, β), where α is the distance to the surface normal of the keypoint, i.e., the radius, and β is the perpendicular distance from the point to the tangent plane:

$$\alpha = \sqrt{\|\mathbf{x} - \mathbf{p}\|^2 - (\mathbf{n} \cdot (\mathbf{x} - \mathbf{p}))^2}, \tag{1}$$

$$\beta = \mathbf{n} \cdot (\mathbf{x} - \mathbf{p}), \tag{2}$$

where $\mathbf{n}$ is the surface normal for keypoint $\mathbf{p}$. Every spin-image bin counts, for a given neighborhood of points around the keypoint, the number of neighbors that fall in a given range of α and β. The procedure is illustrated in Fig. 5c and d. To compute a spin-image, the following parameters must be specified:[8]

– Image width (IW): Defines the resolution of the spin-image, which will be $W + 1$ bins in the radius dimension and $2W + 1$ bin in the distance dimension.

[7] http://www.pointclouds.org/documentation/tutorials/cluster_extraction.php

[8] http://pointclouds.org

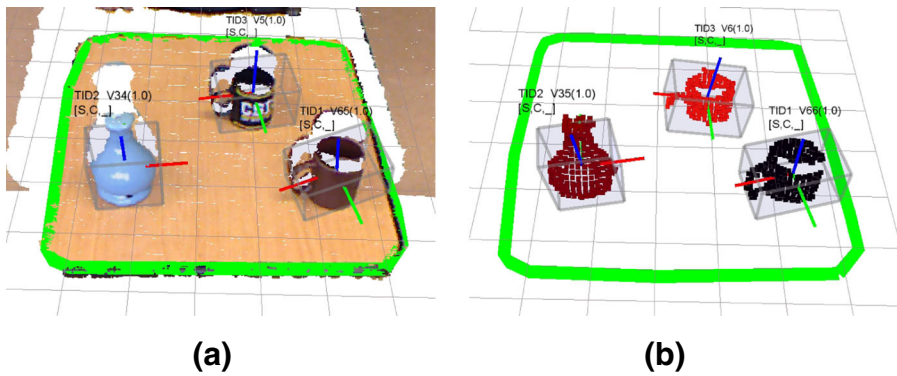

Fig. 4 An example of the object detector module: **a** original point cloud; **b** segmented point clouds. The *red*, *green* and *blue* lines represent the local reference frame of the objects

– Support length (SL): Determines the amount of space swept out by a spin-image, which will have a radius of L and height of $2L$.
– Support angle (A): Maximum angle between the surface normal at the keypoint and the surface normal in other points to be included as neighbors.

In this work, based on off-line experiments described in Section 6.1, we use the following spin-image parameters: $SL = 0.1m$, $A = \pi/2$ and $IW = 8$.

5 Object Conceptualization and Recognition

The previous section addressed the extraction of 3D shape features. In this section, we describe how those features are used both for conceptualizing and recognizing objects. In addition to that, we present the interface that allows a user to label objects. The conceptualization is triggered when the user provides a label to an object.

5.1 User Interface

The open-ended object recognition system will be more flexible if it is able to learn new object categories from its end-users. Once an unknown object is detected, the category of the object must be learned by the system with the help of a human user. The user interface supports the interaction with an instructor for labeling objects. A skeleton tracker library tracks the pose of the instructor over time. This data is given to a gesture recognizer algorithm, which computes the pointing direction. Currently, the pointing direction is assumed as the direction of the right forearm. The User Interface contains a global state of the scene. It

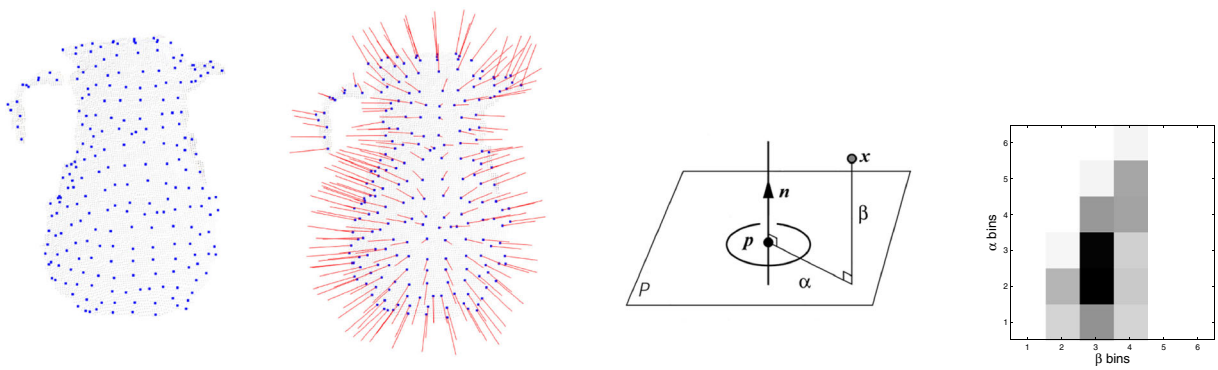

Fig. 5 Computation of the spin-image for a flask: **a** extraction of keypoints; **b** the estimated surface normals; **c** a schematic of how the α and β parameters are computed for a keypoint p; **d** the computed spin image

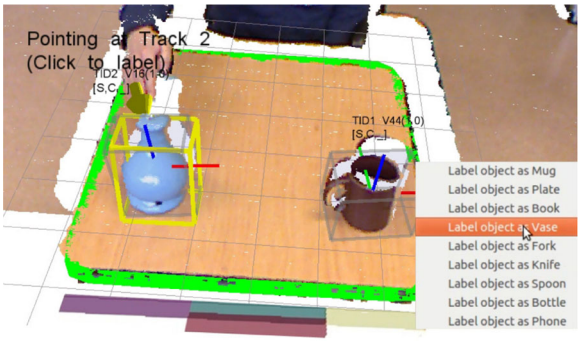

Fig. 6 A 3D visualization of the object labeling mechanism

compares the pointing direction against the position of all detected objects, and decides if an object is being pointed at by the instructor. A menu used for object labeling is provided by the user interface. It appears in front of the table in the 3D visualization of the scene. To produce a valid object category label, the pointing detection must occur simultaneously with the object labeling from the menu. In this way, the false positives possibly produced by the pointing detection mechanism, are ignored. The user interface is shown in Fig. 6.

5.2 Object Conceptualizer

We propose an open-ended 3D object learning and recognition approach based on interactive teaching, in which the set of object categories to be learned is not known in advance. The training instances are extracted from actual experiences of a robot, and thus become gradually available, rather than being available at the beginning of the learning process. The perception system processes point cloud data continuously, and the instructor may choose to point to an object and provide a category label for it. Such situation provides an opportunity to collect a training instance (an experience) for learning. Therefore, an instance-based learning approach is adopted in the current system, i.e. object categories are represented by sets of known instances (Fig. 7). One of the advantages of this approach is to facilitate incremental learning. The object conceptualizer module is activated when the instructor provides a category label for an object. The object conceptualizer retrieves the current object category from memory, as well as a set of features describing the labeled object, and creates a new (or updates the existing) object category by computing the Intra-Category Distance (ICD). To compute the ICD, first a distance between two object views, $\mathbf{U}$ and $\mathbf{V}$, is computed as follows:

$$\mathrm{D}(\mathbf{U},\mathbf{V}) = \frac{\sum_l \min_k \ \mathrm{d}(\mathbf{u}_l, \mathbf{v}_k)}{q}, \tag{3}$$

where, $\mathbf{u}_l$ are the spin-images, $l = 1, \ldots, q$, are the spin-images of the object view $\mathbf{U}$, $\mathbf{v}_k$ represents a spin-image of the object view $\mathbf{V}$, and $\mathrm{d}(\cdot)$ is the Euclidean distance. It should be noted that $\mathrm{D}(\cdot)$ is not symmetric (i.e. $\mathrm{D}(\mathbf{U},\mathbf{V}) \neq \mathrm{D}(\mathbf{V},\mathbf{U})$). Finally, the Intra-Category Distance is computed based on the following formula:

$$\mathrm{ICD}(C) = \frac{\sum_{\mathbf{U}\in C} \ \sum_{\mathbf{V}\in C, \mathbf{U}\neq\mathbf{V}} \mathrm{D}\,(\mathbf{U},\mathbf{V})}{n\cdot(n-1)}, \tag{4}$$

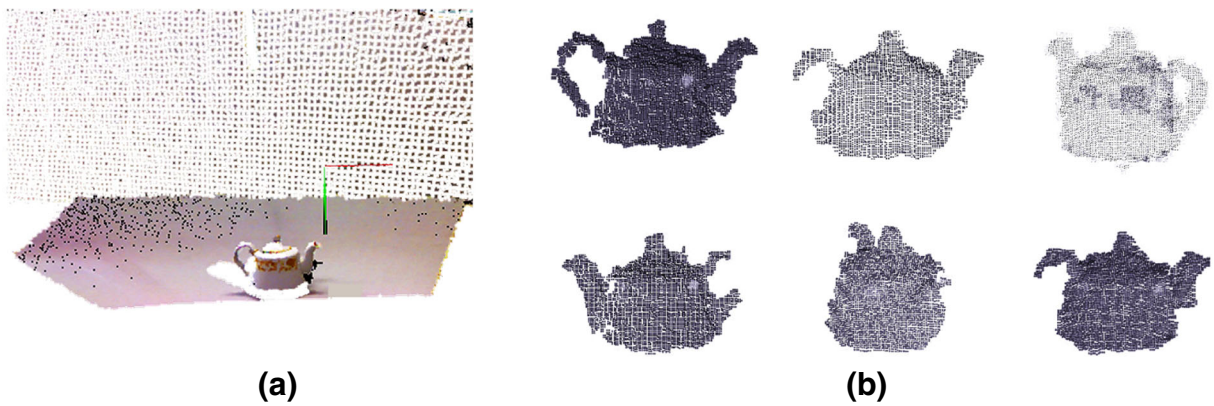

Fig. 7 Object learning: **a** example point cloud of a scene containing a teapot; **b** six views of the segmented teapot from different perspectives

where n is the total number of instances in the category, **U** and **V** are two different known instances of category C.

5.3 Object Recognizer

The object recognizer module is capable of discarding false object candidates as well as objects of unknown categories. It should be noted that the described functionalities for feature extraction and object conceptualization are used in both the learning and recognition situations. The dissimilarity between a target object and an object from the model is estimated based on Eq. 3. The minimum distance between the target object, **T**, and the instances, **O**, of a certain category, C, is considered as the object-category distance:

$$\mathrm{OCD}(\mathbf{T}, C) = \min_{\boldsymbol{O} \in C} \mathrm{D}(\mathbf{T}, \mathbf{O}). \tag{5}$$

Because in some categories, instances are more spread than in others, the object-category distance $\mathrm{OCD}(\mathbf{T}, C)$ should be normalized to prevent misclassification of "unknown" objects into "known" categories. Normalizing the object category distance by the corresponding ICD seems to be a suitable strategy to avoid this problem. Therefore, the normalized distance of target object **T** to the category C, $\mathrm{ND}(\mathbf{T}, C)$, is computed based on following:

$$\mathrm{ND}(\mathbf{T}, C) = \frac{\mathrm{OCD}(\mathbf{T}, C)}{\mathrm{ICD}(C)}. \tag{6}$$

The target object is classified based on the minimum normalized distance to the object. If, for all categories, the normalized distance is larger than a given classification threshold, CT (e.g. CT = 1.0), then, the object is classified as *unknown*:

$$Category(\mathbf{t}) = \begin{cases} unknown, & if \ \min_{C} \mathrm{ND}(\mathbf{T}, C) > \mathrm{CT} \\ \underset{C}{argmin}\ \mathrm{ND}(\mathbf{T}, C), & otherwise \end{cases}. \tag{7}$$

This procedure is illustrated in Fig. 8.

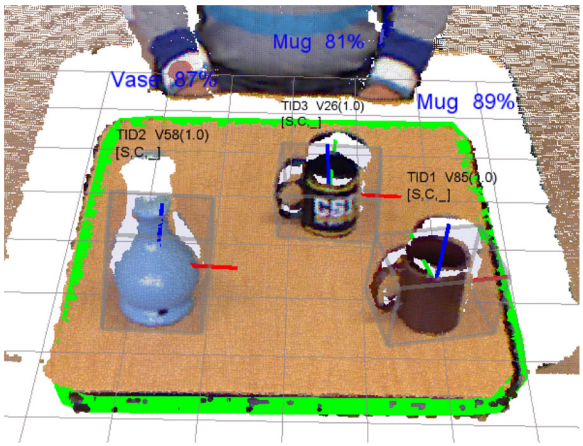

Fig. 8 An example of the 3D object recognition

6 Experimental Results

Three types of experiments were carried out to evaluate the proposed approach. First, an off-line quantitative evaluation for the object recognition system is presented. Secondly, a qualitative analysis of the complete interactive open-ended object recognition system is shown. In this case, a four minute demonstration session is described, where users manipulate objects on a table and interact with the perception system. Finally for performance evaluation in open-ended learning, we use a *simulated teacher* designed to assess the performance of open-ended category learning systems in a systematic and reproducible way [4]. The simulated teacher emulates the interaction of a real human user with the learning system for the purpose of teaching object categories. It successively presents new categories to the learning system, and provides corrective feedback in the case there is a misclassification. All tests were performed with an i7, 2.40GHz processor and 16GB RAM.

6.1 Off-Line Evaluation

Experiments were carried out to evaluate the proposed approach in terms of precision and recall,[9] using real world data. A restaurant object dataset was acquired, which contains 341 views of one instance of 10 categories from different categories (*Bottle*, *Bowl*, *Flask*, *Fork*, *Knife*, *Mug*, *Plate*, *Spoon*, *Teapot*, and *Vase*) and 30 views of false or unknown objects (e.g. points that belong to the instructor's hands). These objects were extracted from 100 views of table-top scenes by running the object detector and storing the segmented

[9] http://en.wikipedia.org/wiki/precision_and_recall

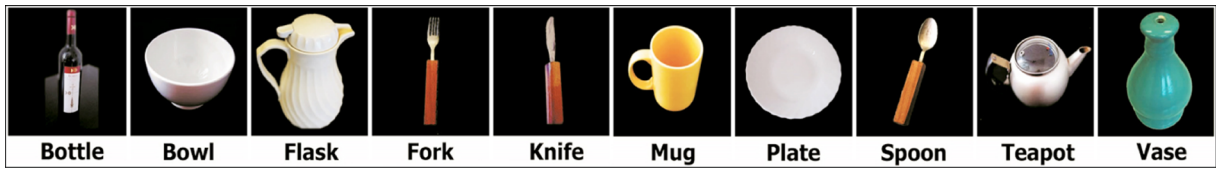

Fig. 9 Objects used for the experiments

point clouds. All segmented point clouds, i.e. the object views, were hand annotated with the respective category labels (Fig. 9).

To examine the performance of different configurations of the proposed approach, a leave-one-out cross validation algorithm was used (see Algorithm 1). In each iteration, one instance of a certain category is used for testing, and the remaining data are used as training data. This procedure is repeated until each instance of all the known categories has been used once as test sample. In case the system predicts a category that is not the true object category, both a false negative (true object category not detected) and a false positive (predicted object category not correct) are accounted for. A total of 96 experiments were performed for different values of four parameters of the system, namely the voxel size (VS) which is related to number of key points extracted from each object view, the image width (IW) and support length (SL) of spin images, and the classification threshold (CT). Results are presented in Table 1. The parameters that obtained the best average F1 score selected as the default system parameters. They are the following: VS = 0.06, IW = 8, SL = 0.1 and CT = 1. The F-measure of the proposed system with the default configurations was 0.81 percent. It shows that the overall performance of the recognition system is promising and that the spin-image descriptor is capable of collecting distinctive traits of the local surface patches of each object. In addition, this configuration displays a good balance between recognition performance, memory usage,and processing speed. Further details of the processing time analysis is available in [19]. The results presented in Sections 6.2 and 6.3 are computed using this configuration.

Table 1 Average object recognition performance for different parameters

Parameters	Values	Average F1
VS	0.02	0.46
	0.04	0.46
	0.06	0.49
	0.08	0.45
IW	4	0.46
	8	0.48
SL	0.025	0.24
	0.050	0.23
	0.075	0.67
	0.100	0.74
CT	0.5	0.42
	0.75	0.49
	1	0.50

6.2 Scenario Based Tests

To show the functionalities of the system, a four minutes long session is used, where several users interacted with the system. During the session, users presented objects to the system and provided the respective category labels. Therefore, throughout this session, the system must be able to detect, conceptualize and recognize new objects, as well as to detect pointing gestures used for labeling them. Initially, the system had no prior knowledge. Figure 10 and the following explanation illustrate the behavior of the main modules of the system, from user and object tracking to learning and recognition. A video of this session is available at: http://youtu.be/XvnF2JMfhvc.

(a) The system works in scenario where a table is in front of the robot and there is no knowledge about any category. The graphical menu in front of the table is the interactive menu that enables teaching new object categories. The instructor puts a *Mug* on the table. Tracking is initialized with track ID 1 (TID 1). The gray bonding box signals the pose of the object as estimated by the tracker. TID 1 is classified as *Unknown* because mugs are not known to the system;

Algorithm 1 Leave-one-out evaluation algorithm

```
Input: DB - a database of object candidates
TP ← 0
FP ← 0
FN ← 0
for each O in DB do
    remove O from DB
    TC ← true category of O
    PC ← category predicted for O
    add O to DB
    if TC = 'unknown' and PC = 'unknown' then
        continue;
    else if TC = PC then
        TP++;
    else if TC = 'unknown' and PC ≠ 'unknown' then
        FP++;
    else if TC ≠ 'unknown' and PC = 'unknown' then
        FN++;
    else if TC ≠ PC then
        FP++;
        FN++;
    end if
end for
```

(b) Instructor labels TID1 as a *Mug*. The system conceptualizes the category;

(c) The *Mug* is correctly classified. The instructor places a *Vase* on the table. Tracking is initialized with TID 2. The *Vase* is unknown to the system; this frame shows that the system is able to detect and track multiple object in the scene. Moreover, it demonstrates that both the tracking and recognition work when the user is holding the objects;

(d) The instructor labels TID 2 as a *Vase*. This labeling is done using a different interaction modality: the instructor points at track ID 2, and labels this object as *Vase*;

(e) The *Vase* is properly recognized. An additional *Mug* is placed at the center of the table. Tracking is initialized with TID 3. This particular *Mug* had not been previously seen, but the system can correctly recognize it, because the Mug category was previously taught. This shows that the system is capable of using prior knowledge to recognize new objects in the scene;

(f) Another instructor arrives; once he sits on front of the robot, he will be considered as the system's

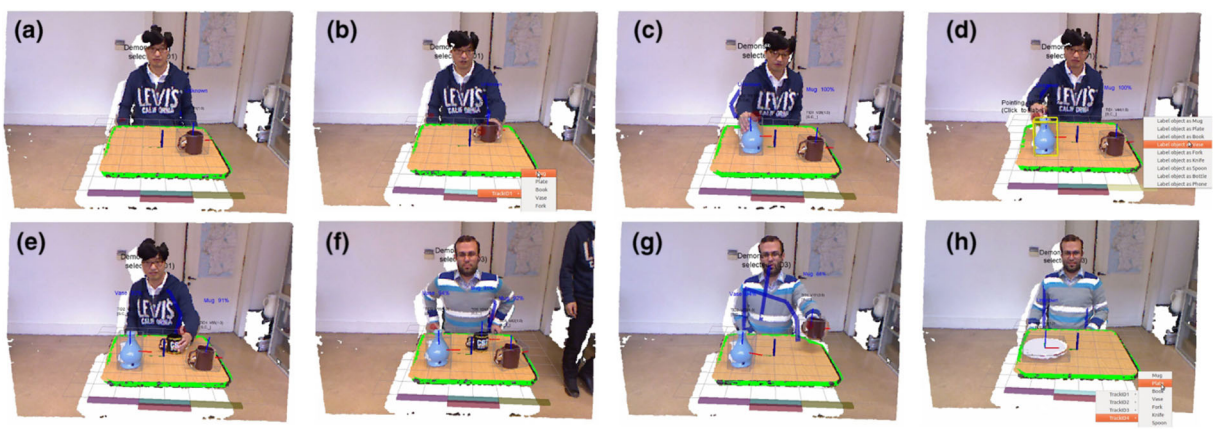

Fig. 10 Sequence of events in the experiment

instructor. This frame shows instructor detection and tracking;

(g) The instructor will remove all objects from the scene; no objects are visible;

(h) A *Plate* enters the scene. It is detected and assigned to TID 4. Because there is no prior knowledge about plates, TID 4 is classified as *Unknown*. TID 4 is labeled as a *Plate* and the system conceptualized the category.

This sequence shows that the proposed architecture is capable of detecting new objects, tracking and recognizing those object in various positions. Moreover, it shows capability of human-robot interaction based on a graphical interface and pointing gesture recognition.

6.3 Open-Ended Evaluation

These tests were conducted using the teaching protocol proposed in [4, 25]. Algorithm 2 shows the pseudo-code for the teaching protocol. A *simulated teacher* was developed to follow the teaching protocol and autonomously interact with the system using *teach*, *ask* and *correct* actions. For each newly taught category, the average sucess of the system should be computed. To do that, the simulated teacher repeatedly picks object views of the currently known categories from a database and presents them to the system for checking whether the system can recognize them. If not, the simulated teacher provides corrective feedback. The database used in these experiments is the one described and used in Section 6.1. Data of false and unknown objects are not used, as they are not required by the teaching protocol.

In the experiments that will be presented, the system begins with zero knowledge and the training instances become gradually available according to the teaching protocol. Therefore, the system incrementally builds object category models. Although the teaching protocol is designed to test the system on categories that have been previously taught, the system can still reply that a given object belongs to an unknown category. Therefore, classification success must be evaluated in terms of precision and recall, leading to the use of F-measure as the measure of classification success. Average success is computed using a sliding window of size $3n$, where n is the number of categories that have already been introduced. If k, the number of iterations since the last time a new category was introduced, is less than $3n$, all results are used. In case the recognition fails to predict the correct category, both a false negative and a false positive are accounted for. Classification success is used as the indicator that a new category can be taught. According to the protocol, the system is ready to learn a new object category when the success is higher that a certain threshold, and at least one instance of every known category has been tested, i.e., $k \geq n$. When an experiment is carried out, learning performance is evaluated based on several measures, namely number of categories learned, number of question / correction iterations, average number of stored instances per category, global success and average classification success. Global success is the percentage of correct classifications made throughout the experiment. Average classification success is the average of all classification success values over all the question / correction iterations. Since the order of introduction of new categories may have a large impact on the performance of the system, five experiments were carried out in which categories were introduced in random sequences (see Table 2). In all experiments, the system learned all 10 categories. Figure 11 shows the performance of the system throughout the first experiment, which was completed in 56 question/correction iterations.

At the beginning of the evaluation procedure two categories (*Mug and Bowl*) are introduced to the system. In question/correction iterations, the classification of the presented instances was correct and thus maximum success (1.0) was achieved in the minimum number of iterations ($k = n = 2$). Then,the

Table 2 Sequences of introduction of categories for the five experiments

Exp#	Introduction Sequence
1	*Mug, Bowl, Spoon, Fork, Knife, Bottle, Teapot, Plate, Flask, Vase*
2	*Spoon, Teapot, Fork, Plate, Vase, Mug, Bowl, Flask, Knife, Bottle*
3	*Bowl, Plate, Fork, Teapot, Bottle, Flask, Mug, Vase, Spoon, Knife*
4	*Vase, Spoon, Fork, Mug, Bottle, Bowl, Teapot, Knife, Plate, Flask*
5	*Bottle, Vase, Flask, Mug, Fork, Spoon, Teapot, Plate, Bowl, Knife*

Algorithm 2 Teaching protocol for performance evaluation [4].

Introduce $Category_1$
$n \leftarrow 1$
repeat
  $n \leftarrow n+1$ ▷ Ready for the next category
  Introduce $Category_n$
  $k \leftarrow 0$
  $c \leftarrow 1$
  repeat
    //question/correction iteration
    Present a previously unseen instance of $Category_c$
    Ask the category of this instance
    If needed, provide corrective feedback
    if $c == n$ **then**
      $c \leftarrow 1$
    else
      $c \leftarrow c+1$
    end if
    $k \leftarrow k+1$
    $s \leftarrow$ success in last $3n$ question/correction iterations
  until ($s >$ success threshold **and** $k >= n$)
    or (user sees no improvement in success))
until (user sees no improvement in success)

simulated teacher introduced a third category (*Spoon*). The simulated teacher presented a randomly selected, previously unseen instance of each of the taught categories, and the system classified them correctly. Therefore, the success remained at 1.0. Next, the simulated teacher presented the *Fork* category. Since the fork has a very similar shape to the spoon, a misclassification occurred, and the success dropped to 0.66. Then, the simulated user provides a corrective feedback in order to increase the recognition success. At iteration 9, this value is above the threshold and a new object category (*Knife*) is presented to the system. When the knife is introduced, success started at 1.0, then dropped to 0.8. The simulated teacher provided corrective feedback and success starts going up again. Afterwards, the simulated teacher presented the

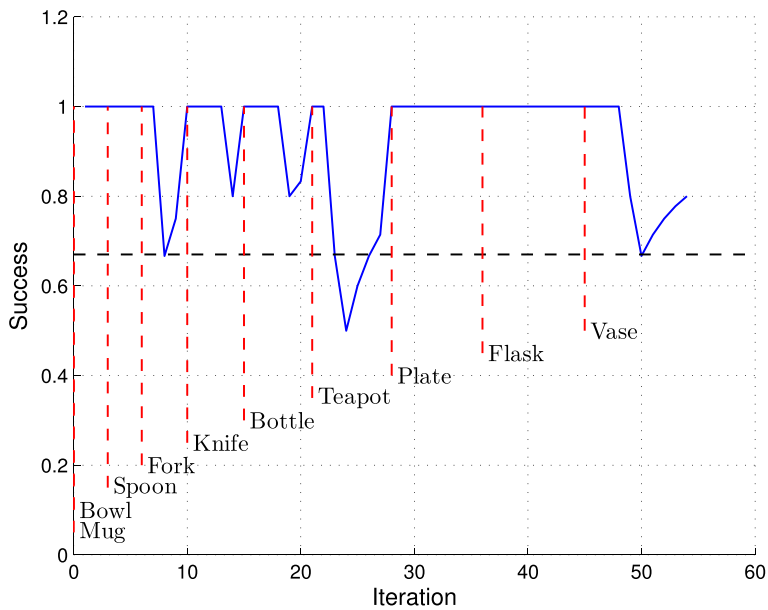

Fig. 11 Evolution of classification success versus number of iterations (Experiment 1)

Bottle and the *Teapot* categories. The success dropped for some iterations and recovered again after receiving corrective feedbacks. For the next two categories (*Plate* and *Flask*), maximum success was obtained, therefore, no corrective feedback was required. Afterwards, the simulated teacher presented the *Vase* category. At 50^{th} iteration, a misclassification occurred, and the success starts going up after receiving corrective feedback. Finally, the experiment finished because no more categories were available to learn. The system successfully learned 10 categories in 54 question/correction iterations. This illustrates the process of acquiring categories in an open-ended fashion using the simulated teacher.

In the additional four experiments, these categories were used again with different introduction sequences, which are reported in Table 2. Figure 12 presents the results obtained for all these experiments, and Table 3 provides a summary.In all experiments the system was able to learn all 10 object categories. By comparing all experiments, it is visible that in the first and second experiments the number of iterations required to learn 10 object categories was greater than other experiments.

In the case of experiment 3, the success remained for the most part above the threshold. Results showed that both evaluation measures (global success and average classification success) for this experiment

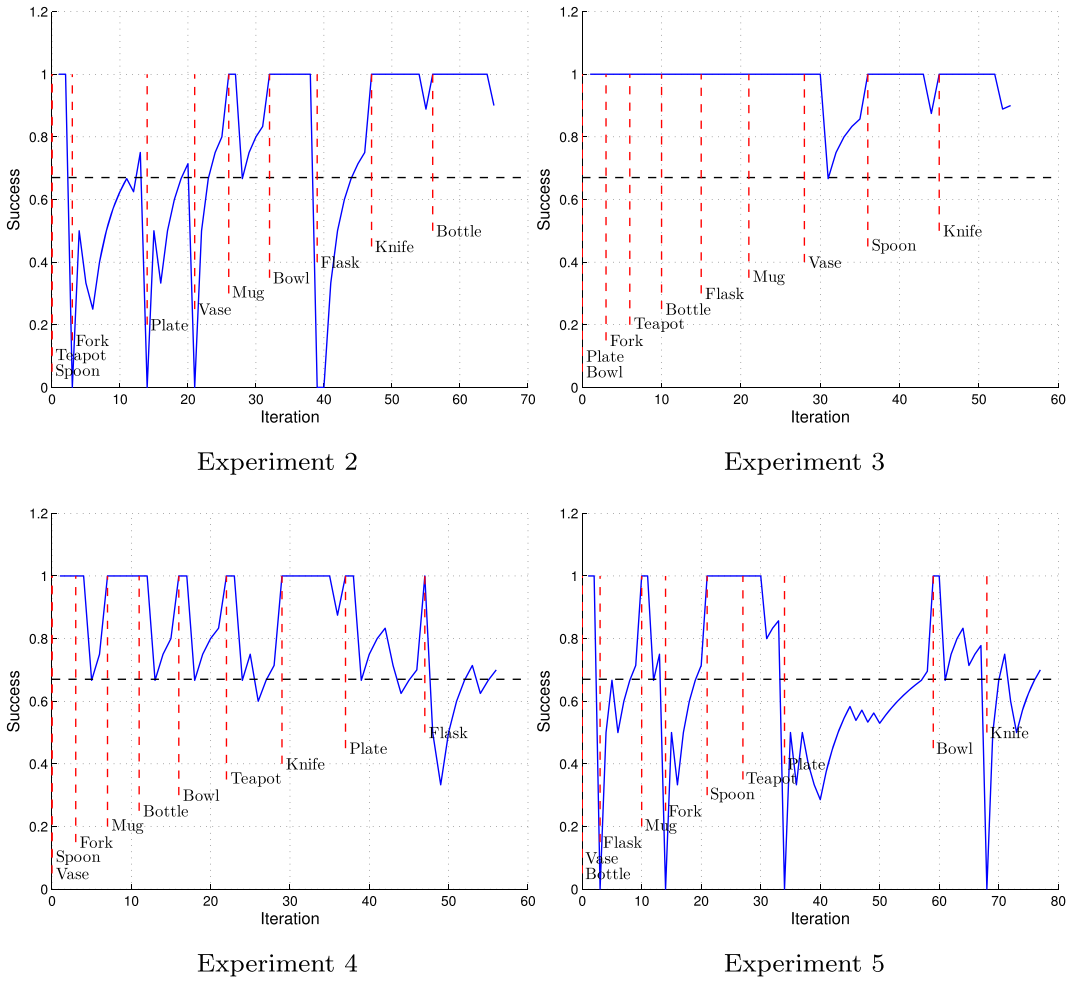

Fig. 12 Evolution of classification success versus number of question/correction iterations for a success threshold of 0.67 (marked by the horizontal line)

Table 3 Summary of experiments[(1)]

Exp#	Threshold	#Iterations	#Categories	#Instances	GS	ACS
1	0.67	54	10	3.7	0.87	0.87
2	0.67	65	10	4.3	0.81	0.84
3	0.67	54	10	3.6	0.94	0.95
4	0.67	56	10	4.2	0.78	0.81
5	0.67	77	10	4.9	0.75	0.80

[(1)]Exp#, experiment number; Threshold, success threshold (s in Algorithm 2); #Iterations, total number of iterations in the experiment; #Categories, total number of categories learned; #Instances, average number of instances per category at the end of the experiment; GS, global success; ACS, average classification success.

are higher than in all other cases. In addition, this experiment and the first one, were the experiments that learned all 10 categories faster (54 iterations, see Table 3). The underlying reason for different performances of these experiments is that categories were introduced to the system in a different order, which has a significant influence on the evolution of the learning performance.

7 Conclusions and Future Work

This paper presented an open-ended 3D object recognition approach based on interactive learning of multiple object categories. In open-ended settings, the set of object categories to be learned is not known in advance, which leads to the fact that training instances must be extracted from on-line robot experiences, rather than being available at the beginning of the learning process. To support open-ended learning, we have developed a system that allows the user to label objects by pointing at them and selecting the corresponding category from a menu interface. This approach was tested on a PR2 robot as shown in Fig. 13. Results showed that the proposed system obtains good precision and recall figures in the off-line evaluation. The on-line evaluation proved that the system can incrementally learn new object categories. For future work, the evaluation of the system using additional table-top scenarios is considered. We would like to investigate the possibility of improving performance using other 3D shape descriptors or more compact representation such as in the bag-of-word approach.

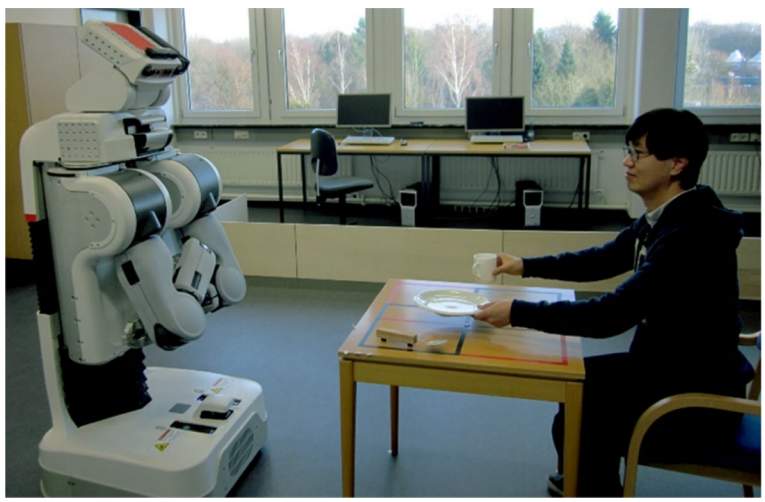

Fig. 13 The proposed system being tested on the PR2 robot

Acknowledgments This research is partially funded by the Robustness by Autonomous Competence Enhancement (RACE) EU project FP7- 287752, COMPETE/FEDER and by National Funds through FCT in the context of the project FCOMP-01-0124-FEDER-022682 (FCT ref. PEst-C/EEI/UI0127/2014). The first author is supported by FCT under grant SFRH/BD/94183/2013. We would like to thank the other RACE project partners for their efforts in the integration and the demonstrations, and especially to the Technical Aspects of Multimodal Systems (TAMS) group, University of Hamburg, for making the PR2 robot available to the project. We also thank the anonymous reviewers for many helpful comments.

References

1. Andreopoulos, A., Tsotsos, J.K.: 50 years of object recognition: Directions forward. Comp. Vision Image Underst. **117**(8), 827–891 (2013)
2. Beis, J.S., Lowe, D.G.: Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), 1997. IEEE Computer Society, Washington, DC, USA (1997)
3. Campbell, R.J., Flynn, P.J.: A survey of free-form object representation and recognition techniques. Comp. Vision Image Underst. **81**(2), 166–210 (2001)
4. Chauhan, A., Lopes, L.S.: Using spoken words to guide open-ended category formation. Cogn. Process. **12**(4), 341–354 (2011)
5. Collet Romea, A., Berenson, D., Srinivasa, S., Ferguson, D.: Object recognition and full pose registration from a single image for robotic manipulation. In: IEEE International Conference on Robotics and Automation, (ICRA 2009) (2009)
6. Dinh, H., Kropac, S.: Multi-resolution spin-images. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 863–870 (2006)
7. Filipe, S., Alexandre, L.A.: A comparative evaluation of 3d keypoint detectors in a rgb-d object dataset. In: 9th International Conference on Computer Vision Theory and Applications. Lisbon, Portugal (2014)
8. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
9. Hertzberg, J., Zhang, J., Zhang, L., Rockel, S., Neumann, B., Lehmann, J., Dubba, K., Cohn, A.G., Saffiotti, A., Pecora, F., Mansouri, M., Konečný, Š., Günther, M., Stock, S., Lopes, L.S., Oliveira, M., Lim, G.H., Kasaei, H., Mokhtari, V., Hotz, L., Bohlken, W.: The race project. KI - Künstliche Intelligenz, pp. 297–304 (2014). doi:10.1007/s13218-014-0327-y
10. Islam, M., Jahan, F., Min, J.H., hwan Baek, J.: Object classification based on visual and extended features for video surveillance application. In: Control Conference (ASCC 2011), 8th Asian, pp. 1398–1401 (2011)
11. Jeong, S., Lee, M.: Adaptive object recognition model using incremental feature representation and hierarchical classification. Neural Netw. **25**, 130–140 (2012)
12. Johnson, A., Hebert, M.: Using spin images for efficient object recognition in cluttered 3d scenes. IEEE Trans. Pattern. Anal. Mach. Intell. **21**(5), 433–449 (1999)
13. Kasaei, H., Oliveira, M.R., Lim, G.H., Lopes, L.S., Tomé, A.M.: An interactive open-ended learning approach for 3d object recognition. In: Proceedings of the 2014 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC) (2014)
14. Kirstein, S., Wersing, H., Gross, H.M., Körner, E.: A lifelong learning vector quantization approach for interactive learning of multiple categories. Neural Netw. **28**, 90–105 (2012)
15. Kootstra, G., Ypma, J., De Boer, B.: Active exploration and keypoint clustering for object recognition. In: IEEE International Conference on Robotics and Automation, (ICRA 2008), pp. 1005–1010 (2008)
16. Liu, Y., Zha, H., Qin, H.: Shape topics: A compact representation and new algorithms for 3d partial shape retrieval. In: Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, vol. 2, pp. 2025–2032 (2006)
17. Martinez Torres, M., Collet Romea, A., Srinivasa, S.: Moped: A scalable and low latency object recognition and pose estimation system. In: IEEE International Conference on Robotics and Automation, (ICRA 2010) (2010)
18. Mian, A., Bennamoun, M., Owens, R.: On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. Int. J. Comput. Vis. **89**(2–3), 348–361 (2010)
19. Oliveira, M., Lim, G.H., Seabra Lopes, L., Kasaei, H., Tome, A., Chauhan, A.: A perceptual memory system for grounding semantic representations in intelligent service robots. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE (2014)
20. Ozawa, S., Toh, S.L., Abe, S., Pang, S., Kasabov, N.: Incremental learning of feature space and classifier for face recognition. Neural Netw. **18**(5–6), 575–584 (2005)
21. Rockel, S., Neumann, B., Zhang, J., Dubba, S.K.R., Cohn, A.G., Konecny, S., Mansouri, M., Pecora, F., Saffiotti, A., Günther, M., et al.: An ontology-based multi-level robot architecture for learning from experiences. In: Proceedings of the AAAI Spring Symposium: Designing Intelligent Robots (2013)
22. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. Int. J. Comput. Vis. **77**(1-3), 125–141 (2008)
23. Rusu, R.B., Bradski, G., Thibaux, R., Hsu, J.: Fast 3d recognition and pose using the viewpoint feature histogram. In: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pp. 2155–2162. IEEE (2010)
24. Schulz, D., Burgard, W., Fox, D., Cremers, A.: Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In: IEEE International Conference on Robotics and Automation, (ICRA 2001), vol. 2, pp. 1665–1670 (2001)

25. Seabra Lopes, L., Chauhan, A.: How many words can my robot learn? An approach and experiments with one-class learning. Interact. Stud. **8**(1), 53–81 (2007)
26. Seabra Lopes, L., Chauhan, A.: Open-ended category learning for language acquisition. Connect. Sci **20**(4), 277–297 (2008)
27. Takamuku, S., Hosoda, K., Asada, M.: Shaking eases object category acquisition: Experiments with a robot arm. In: Proceedings of the Seventh International Conference on Epigenetic Robotics (2007)
28. Tombari, F.: Di Stefano, L.: Object recognition in 3d scenes with occlusions and clutter by hough voting. In: 4th Pacific-Rim Symposium on Image and Video Technology (PSIVT 2010), pp. 349–355 (2010)
29. Wohlkinger, W., Vincze, M.: Shape-based depth image to 3d model matching and classification with inter-view similarity. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011), pp. 4865–4870 (2011)
30. Yeh, T., Darrell, T.: Dynamic visual category learning. In: IEEE Conference on Computer Vision and Pattern Recognition, (CVPR 2008), pp. 1–8 (2008)