A Multimodal Dataset for Interactive and Incremental Learning of Object Models

Pablo Azagra, Yoan Mollard, Florian Golemo, Ana C Murillo, Manuel Lopes, Javier C Civera

A Multimodal Dataset for Interactive and Incremental Learning of Object Models

Pablo Azagra[1], Yoan Mollard[2], Florian Golemo[2], Ana C. Murillo[1], Manuel Lopes[3], Javier Civera[1]

***Abstract*— This work presents an incremental object learning framework oriented to human-robot assistance and interaction. To learn new object models from interactions with a human user, the robot needs to be able to perform multiple recognition tasks: (a) recognize the type of interaction, (b) segment regions of interest from acquired data, and (c) learn and recognize object models. The contributions on this work are focused on the recognition modules of this human-robot interactive framework. First, we illustrate the advantages of multimodal data over camera-only datasets. We present an approach that recognizes the user interaction by combining simple image and language features. Second, we propose an incremental approach to learn visual object models, which is shown to achieve comparable performance to a typical offline-trained system. We utilize two public datasets, one of them presented and released in this work. This dataset contains synchronized recordings from user speech and three cameras mounted on a robot, which captured the user teaching object names to the robot.**

I. Introduction

We are witnessing widespread adoption of service robotics in multiple aspects of our daily life and activities, such as household assistance devices or autonomous cars. One of the key aspects in service robotics is a comfortable and intuitive human-robot interaction. For such, it is essential to learn world models, affordances, and capabilities from the user's knowledge and behavior.

Learning systems can be trained offline, using all the data available at a specific moment, or incrementally, by augmenting and updating the learned model as new samples of data become available. Our work is focused on this latter option, for two reasons. Firstly, although offline learning has shown impressive performance, specially since the recent deep learning wave, it requires copious amounts of data. This amount of data is not available for all the relevant scenarios of human-robot interaction and other strategies are required. Secondly, after an initial learning phase, a service robot environment might change abruptly, which would require re-learning models from scratch. An incremental learning scheme is essential to deal with those changes.

From the robot perspective, a framework for interactive object learning from a human user should contain multiple modules: 1) recognizing the action/interaction that the user is performing; 2) segmenting the image regions relevant to the current object of interest; 3) recognizing the class of the object of interest and incrementally learning a visual model for each object class. Our current work is focused on the recognition steps (user interaction type and object classes).

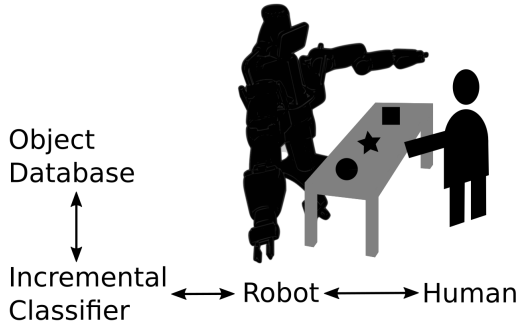


Fig. 1: Schematic for interactive incremental learning. Human and robot interact over objects on a table. Through the communication the robot learns to name and recognize new objects.

This work presents several contributions towards a long-term goal of intelligent systems capable of assisting human users in daily tasks: (a) We present an interaction framework and setup based on language and vision for the incremental learning of object models. Fig. 1 shows a schematic of this framework. (b) We release a dataset, acquired on the presented setup, for interactive and incremental learning. It contains synchronized audio, multi-camera videos (three cameras), and depth information (two of the cameras are RGB-d sensors). To our knowledge, this is the first available dataset that offers user interaction data from the robot's perspective with multi-camera and microphone synchronized. (c) We present an illustrative example of the benefits of multimodal data to recognize types of user interaction with the robot, while the user is teaching different object classes to the robot. (d) We propose an incremental visual learning scheme for object models, that achieves results comparable to a common offline learning approach, within just a few iterations.

II. Related work

In recent years, significant advances have been made in the field of incremental learning. Incremental learning in general is perfectly suited to robotics, as the data arrives sequentially and the robot needs to keep the best model up to date at real time (e.g., inverse dynamics incremental learning in [10]). More specifically, incremental learning is highly valuable in scenarios where human-machine interaction is required.

Our work is oriented towards this direction. There are plenty of applications where a service robot assists a human

[1] DIIS-I3A. University of Zaragoza, Spain.
[2] INRIA Bordeaux Sud-Ouest, France
[3] INESC-ID, Instituto Superior Técnico, Univ. de Lisboa, Portugal

user to perform certain tasks [20], [4], [9]. Through the interaction, the robot incrementally learns and improves the models required for the assistance. Bohg et al. [5] presented a very recent survey on interactive perception and how it can be leveraged for robotic actions, with specific references for interactive object modeling.

Frequently in related work, the robot interacts directly with the scene, e.g. grasping and moving an object, to build an incremental object model [14], [12], [15], [23], [7]. Our approach is complementary to these works, as human interaction is needed in real scenarios, e.g. if the object is out of reach of the robot. Very related to our work, Pascuale et al. [19] uses Convolutional Neural Network-based features and Support Vector Machine classification for visual recognition. Training data used there consists of egocentric images where a human presents an object in front of the robot. Camoriano et al. [6] harnessed that data (vision-only data and user interactions consisting always of a user showing the objects to the robot) and presents a variation of Regularized Least Squares for incremental object recognition. The contribution of our proposal, over these works is the use of multimodal data and different types of user interactions, such as pointing interactions, aiming at a more natural human-robot interaction.

In mobile robotics, we find multiple examples that propose how to incrementally adapt environment visual models as the robot moves. These approaches are often based on Gaussian mixture models that can be easily updated and maintained to recognize regions of interest for the robot [8], [21]. Other works, such as Angeli et al. [3], present an incremental method to build a model to recognize visual loop-closures.

The relevance of interactive and incremental recognition extends beyond the field of robotics. Yao et al. [26] proposed an incremental learning method, that continually updates an object detector and detection threshold, as the user interactively corrects annotations proposed by the system. Kuznetsova et al. [16] investigated incremental learning for object recognition in videos. Lee et al. [18] presented a SIFT-vocabulary that constructs an incremental graph from a single image.

The main problem we study in this work is how to learn an incremental model for object recognition from the robot. Object recognition is a traditional research area in computer vision and robotics, and the literature is densely populated with public datasets. For example [17], [24] are two well-known examples of datasets targeting object recognition from *RGB-D* images. However, most of the existing datasets focus on offline visual learning. Interactive, multi-sensor, and multimodal datasets are scarcer. There are multiple aspects that should be considered on a dataset targeted for interactive learning. In particular, we focus on the expected manner that the data will be presented and multimodal data available.

In a realistic human-robot interactive learning scenario, the training data is expected to be presented in a very different way than the previously mentioned datasets. The data is expected to be shown by a human actor through different ways of interaction. The data is expected to be seen from the point of view of the robot. Recognizing a pedestrian from a close-up view from service robot is immensely different from performing the same task with the raw video data from a distant wide-angle surveillance camera.

Vatakis et al. [25] shows multimodal recording approach similar to ours, but the purpose of their dataset was to capture the reactions of users to stimuli with objects or images in a screen. Datasets like [13] or [11] capture human-robot interaction from a third-person point of view (POV). This is useful in some cases, but since we are working with service robots, information must be taken from the onboard sensors. Additionally, many datasets lack additional sensor data that is easy to find in human-robot interactive scenarios like speech from the user. As detailed in next section, our released dataset is focused not only on capturing the user's expression, but also on capturing the scene information (hands, arms, desk, objects, etc.) related to the object classes being taught to the robot.

III. Multimodal human-robot interaction dataset

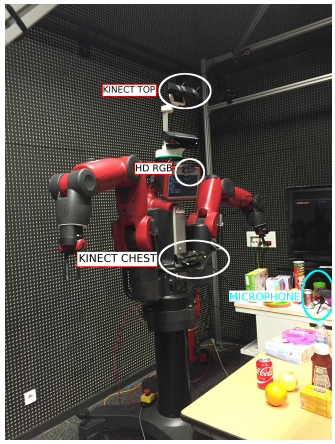


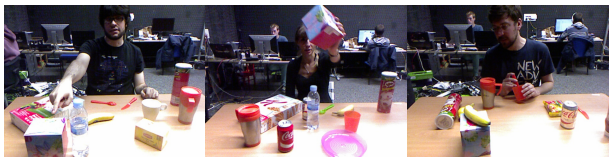
Fig. 2: Image of the Baxter robot used to acquired the dataset. The three camera positions are highlighted.

The dataset[1] contains recordings of several users teaching different object classes to the robot. It contains synchronized data from two Microsoft Kinect v1.0, a 1280×720 *RGB* camera, and a microphone. The cameras are mounted so that the user and the table on which the interaction occurs, are perfectly covered. The first, torso-mounted *RGB-D* Kinect is focused on the frontal interaction with the robot, the second one (head-mounted) gives an aerial view of the table and the *HD-RGB* camera is focused on the user's face. Figure 2 shows the placement of the cameras in the Baxter robot used for the acquisition.

The types of interactions captured in the dataset reflect the most common ways users show or teach objects to the robot: *Point*, *Show*, and *Speak*. The interaction *Point* captures a user pointing at an object and calling out its name to the robot. The interaction *Show* describes a user grabbing an object and

[1] http://robots.unizar.es/IGLUdataset/

bringing it approximately in front of the robot's torso camera while announcing the object's name. The interaction *Speak* captures a user describing to the robot where a certain object is placed. Figure 3 shows an example of each of these three types of interactions.



(a) Pointing (b) Showing (c) Speaking

Fig. 3: Three types of user interaction. In the respective cases the user would say for example (a) "This is a box", while pointing to the box. (b) "This is a box", while holding the box. (c) "The box is next to the chips and has a banana on top."

Table I contains a summary of the contents of the dataset. We recorded 10 users, each of them performing 10 object interactions for each of the 3 tasks (point, show, speak) for a total of 300 recordings. The users were allowed to freely choose which objects they wanted to explain out of a pool of 22 objects on the table. They were given unspecific instructions on the interactions, meaning no exact phrasing. As a result, there is a natural variation in language usage between speakers. Figure 4 shows some examples of the dataset recordings.

Users	10	
Interaction Types (Actions)	3	*Point, Show, Speak*
Interaction per User	30	10 of each type. 1 Object per interaction.
Objects	22	*Apple, Banana, Bottle, Bowl, Cereal Box, Coke, Diet Coke, Ketchup, Kleenex, Knife, Lemon, Lime, Mug, Noodles, Orange, Plate, Spoon, Tall Mug, Tea Box, Vase*

TABLE I: Overview over the parameters which were set to record the dataset.

IV. Learning from multimodal interactions.

One of the main goals in our research is to exploit multimodal data for interactive learning, with a focus on natural exchange between robot and human. This section describes an illustrative experiment which shows the advantages of utilizing more than one modality of data. As we can see in [1], combining language and image data can significantly boost the user action recognition. In our case action recognition is not the goal of the experiment, but only an initial step to facilitate the incremental object learning by the robot.

Our dataset includes three types of interaction: *Point*, *Show*, and *Speak*. A rough classification of the interaction can simplify and optimize the object segmentation step, in which the robot needs to extract the object's visual features. For example, if the user is pointing to an object, the object is very likely to be within a very restricted image region that we can estimate. If the user is grabbing an object and showing it to the robot, our system can assume that the object is held by the user's hand. In this case, it could predict that the hand is likely to occlude parts of the object.

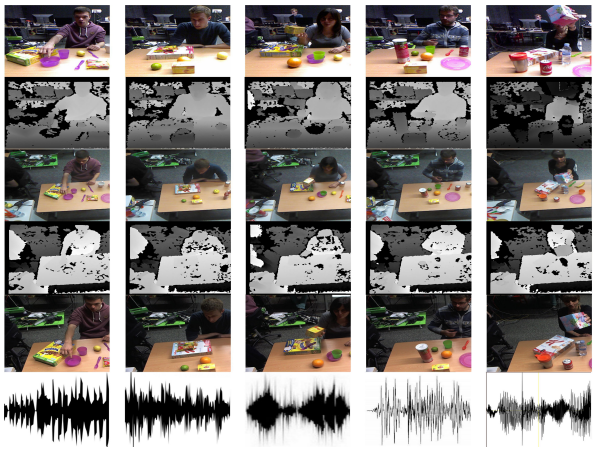


Fig. 4: Examples from the Human-robot interaction dataset. Each column shows the same event captured from the 4 sensors.

A. Multimodal interaction recognition

This experiment aims to recognize the three types of interactions that occur in the dataset (*Point*, *Show*, and *Speak*) from language and visual cues. We are seeking a simple prefilter for the incremental object learning, in order to avoid costly spatio-temporal video analysis, and propose a low-cost individual frame classifier. Firstly, notice in Figure 6 the difficulty of the interaction recognition problem from a single frame. Classifying these 4 frames into *Point*, *Show*, or *Speak* can be tricky even for a human.

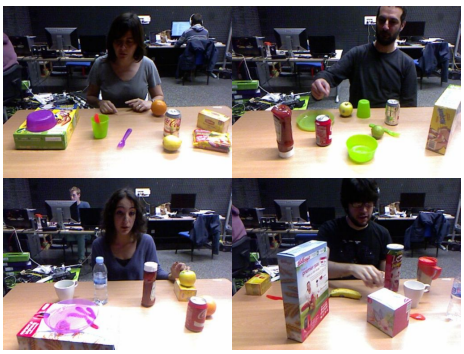


Fig. 5: Sample frames from user interactions with the robot. It is hard to distinguish from a single frame if the user is pointing at an object, about to pick it up, or just narrating.

Visual data features: Our image descriptor is computed as follows. First, we segment the image into SLIC [2] superpixels. We classify each superpixel as skin/not skin using color and depth and fuse the adjacent skin superpixels

into blobs. Afterwards we divide the image into a 5×5 grid, with the descriptor being a histogram summing the skin votes of the blobs per each image cell.

Language data features: Our language feature is the first word of each user sentence, which is 1) "*that*" when the user is pointing to something far; 2) "*this*" when the user is pointing to something close or showing something to the robot; 3) any other word if the user is just describing, saying, something to the robot. Figure 6 shows the language feature distribution for each user and interaction. Notice that the feature is not discriminative at all for the classes *Point* and *Show*, and therefore training a classifier with only this feature is not an option because the precision on those classes will be near random.

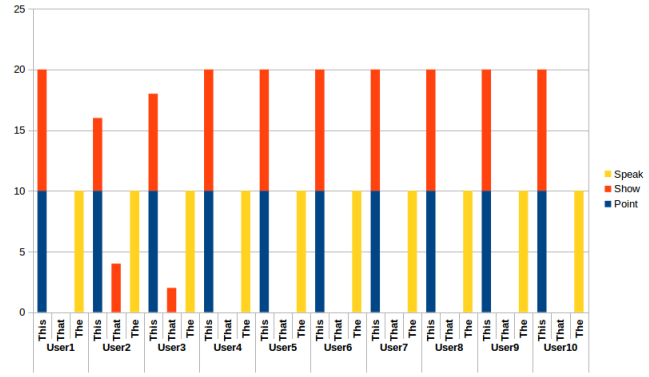


Fig. 6: Stacked graphic showing the language feature occurrences during the three types of interaction from all users.

Interaction classifier: We train a RBF-SVM classifier first using only visual data and then using the combination vision-language. We report the confusion matrix (10-fold validation) for both cases –Table II is vision-only and Table III is vision-language. Notice the boost in the performance for the multimodal descriptor, in particular for the class *Speak* that is ambiguous for the visual descriptor.

	Point	**Show**	**Speak**
Point	**27,74%**	4,71%	5,63%
Show	16,96%	**2,87%**	2,48%
Speak	21,97%	7,92%	**9,72%**

TABLE II: Confusion matrix for interaction recognition using visual data only.

	Point	**Show**	**Speak**
Point	**29,39%**	10,36%	0,00%
Show	13,13%	**6,63%**	0,00%
Speak	0,32%	0,00%	**40,17%**

TABLE III: Confusion matrix for interaction recognition using visual and speech data.

V. Incremental Learning of Object Models

We work towards a system where a robot can learn object models incrementally, while maintaining a limited amount of data stored. Our approach works with object views which are windows of images containing the objects. When the system gets a new object view, it attempts to assign a label from its database of currently know objects (if any). Based on the confidence of this assignment, the robot will either merge this information into the corresponding object cluster information or ask the user if the answer is correct. If the user confirms the label is correct, this new view of the object is merged to its model in the database. If the assigned label is wrong, the user will provide the label name of this new object and it will be incorporated into the database.

Our object model database consists of a set of representative descriptors for each object, described in sec. V-A. Each of these representative descriptors can be seen as the centroid of a cluster in the database. When a new object is added, the descriptor of that first object view is directly used as the seed centroid of a new cluster.

A. Object View Description

Our system is designed to run on robotic platforms, where computational performance is limited. Therefore, we propose to use descriptors that are reasonably small and fast to compute, but still able to discriminate among the common objects the robot will have to interact with. Our system uses two kind of descriptors to represent the image region enclosing the object we aim to learn:

BoW Histogram: This descriptor consists of a histogram of the frequency of occurrences of a set of visual words built from commonly used local image features. In particular, we use ORB [22] features, since they provide a good compromise between efficiency and amount of key points detected.

The visual words, or vocabulary, are obtained as a result of clustering all the point features extracted on a large set of images from a public object dataset [17]. We compute 1000 clusters from more than 2 million features extracted from around 12000 images. These images contain both clean images of all object classes in the dataset and test scene images, with objects and clutter.

To build the descriptor d_{BoW} of a new image i, we extract point features, find the closest word to each of them and calculate d_{BoW} as a 1000-bin histogram of the frequency of occurrence t_w of each word in the image as:

$$\begin{aligned} d_{\text{BoW}} &= [t_1, ..., t_w, ...t_{1000}] \\ t_w &= \frac{n_{\text{wi}}}{n_i}, \end{aligned} \tag{1}$$

where n_{wi} is the number of occurrences of word w in image i and n_i is the total number of words in image i.

Color Histogram: This descriptor d_{RGB} provides information about the color distribution in the image region described. We compute three normalized 8-bin histograms

over the color pixel information on the region. There is one histogram, H_c, for each RGB channel:

$$d_{\text{RGB}} = [H_r \; H_g \; H_b] \tag{2}$$

Our experiments evaluate the results, by using only one of these descriptors or by combining them. Since the image region representing an object can be small, there could be none or only very few point features present. Consequently not enough information would be obtain from them, as can be seen for example in Figure 7. Therefore, although point features are usually more discriminative than color histograms, as shown in section VI-C, the best option is the combination of both descriptors, which we will refer to as d_{ALL}.

Fig. 7: Sample new objects views provided by the user to the robot. These objects segments occupy small image regions where there are often not enough point features found.

B. Classify and Learn from a New Object View.

Our incremental system is inspired by incremental clustering ideas. Each cluster in our model represents a representative subset of the descriptors seen so far for a certain class. As we get new samples, existing clusters evolve and update their centroids (representative descriptors) and new clusters are created for new objects. The total number of classes (N_Class) is not limited but, in order to avoid unlimited growing, the subset for each class is limited by a predefined size **S**. The algorithms proposed to build this model (*training*), and to classify the object (*recognition*) are shown in pseudo-code in algorithm 1 and algorithm 2 respectively.

Incremental training: Given a new view object v and a label l, first the descriptor d^v is calculated. A new cluster is created with d^v as centroid and l as label associated. If the label l doesn't exist in the database L, l is added to the database. If l has reached the maximum number of clusters associated **S**, the algorithm finds the clusters with minimum distance between them of this class and merges them. This method limits the amount of overlap that can occur in classes and maintain real data in the model.

C_l is the set of clusters corresponding to label l. The distance used to compare two clusters D_B is the Bhattacharya distance between their centroid descriptors.

$$(\hat{x}, \hat{y}) = argmin_{x,y}\{D_B(C_l[x], C_l[y])\} \ni x \neq y \tag{3}$$

Clusters $C[\hat{x}]$ and $C[\hat{y}]$ are merged into one cluster $C[z]$ with L associated and the centroid is randomly chosen from one of them.

```
Data: v, l
d^v=Calculate_descriptor(v);
if N_Class == 0 then
    L.add_class(l);
    C.create_cluster(l,d^v);
else
    C.create_cluster(l,d^v);
    if l is not in L then
        L.add_class(l);
    else
        if len(C_l) > S then
            distance_min = inf;
            for each x in C_l do
                for each y in C_l do
                    if x ≠ y then
                        d = D_B(C[x],C[y]);
                        if d < distance_min then
                            distance_min = d;
                            x̂ = x;
                            ŷ = y;
                        end
                    end
                end
            end
            C.Merge(C[x̂],C[ŷ]);
        end
    end
end
```

Algorithm 1: Training Algorithm

```
Data: v
Output: l,Confidence
d^v=Calculate_descriptor(v);
if N_Class == 0 then
    return None,False;
else
    distance_min = inf;
    for each x in C do
        d=D_B(d^v,C[x]);
        if d < distance_min then
            distance_min = d;
            x̂ = x;
        end
    end
    if distance_min < Th then
        confidence = True;
    else
        confidence = False;
    end
    return L[x̂], Confidence;
end
```

Algorithm 2: Recognition Algorithm

Recognition of a new view: Our recognition step classifies new object view v into the existing classes following a simple nearest neighbor approach. It computes a distance from the descriptor d^v of this new view to the centroid of each model cluster. The distance D_B is the Bhattacharya distance between v and the model centroids. C is the set of the current model clusters, and $C[x]$ represents the centroid of cluster x.

$$\hat{x} = argmin_x\{D_B(v,\, C[x])\} \tag{4}$$

Since each existing cluster has an object label assigned, let us represent it as $L[x]$, the new view will be classified as class $L[\hat{x}]$.

To take into account the confidence in the classification result, we establish a threshold, Th, on the distance to nearest

% train. data	20%	30%	40%	50%	60%	70%	80%	90%
Offline	746	1119	1492	1865	2238	2611	2984	3357
Incremental	684	873	968	1005	1014	1020	1020	1020

TABLE IV: Database size for each method using different % of training data from the Washington dataset.

neighbor found on the database, i.e., $D_B(v, C[\hat{x}])$. If it is above this threshold, the system asks the user about the label for this object. Th is experimentally set as we explained in section VI-A.

VI. Experimental Results

The goal of the experiments described in this section is to show the performance of our incremental learning algorithm for object recognition.

A. Experimental Setup

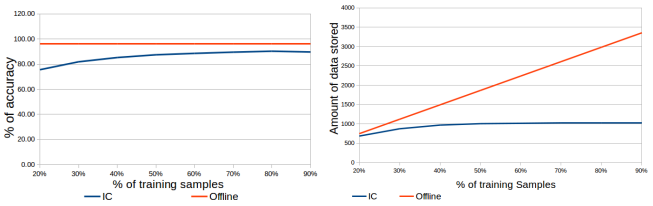
The baseline considered for these experiments is an offline trained nearest-neighbor classifier that uses all the training data. We run 10-fold cross-validation for all of our experiments (both for the baseline and our incremental approach), making sure that there are at least two images of the same class in each fold. The evaluation metrics we consider are the accuracy (defined as the true positives over the total test samples) and the amount of data stored.

The parameter values for our approach, which are used in all the experiments, are as follows. The maximum number of clusters allowed for the model of one class is $S = 20$. The threshold, Th, set to decide when the descriptor distance is low enough for the robot to take a decision, i.e., assign a class to an object view, without asking the user (as explained in section V-B), is set to 0.1 for d_{RGB}, 0.7 for d_{BoW} and 0.7 for d_{ALL}. These thresholds were set in such a way that they were providing a 99% of accuracy exceeded in the classification performed.

The algorithms were developed in Python using OpenCV and Sklearn libraries, and the experiments were run on a PC with Intel® Core™i7-6700 CPU at 3.40GHz, 32GB RAM.

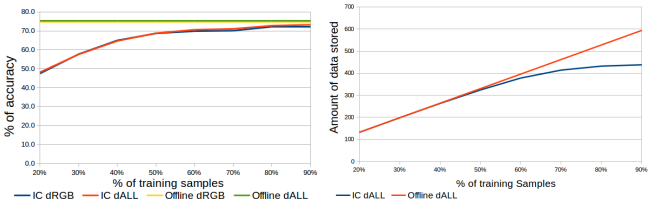
B. Results on the Washington Dataset

In this first experiment, we evaluate our approach, with respect to the offline baseline approach, using data from the Washington dataset [17]. This dataset contains small and clean images of objects. This is in contrast to our dataset, which is noisier and prone to occlusions. We use the two descriptors proposed combined (d_{ALL}). Figure 8 and Table IV summarize the results. In Figure 8(a) we can see that our system's performance quickly reaches 90% accuracy, close to the baseline accuracy of 96%. Interestingly, Figure 8(b) shows that we require only to store one third of the data, compared to baseline, to achieve this result. Additionally, Table IV shows that our data usage converges to around 1000 samples, whereas the baseline linearly increases. This shows that our system maintains a stable precision without having an increase in the amount of data stored.



(a) (b)

Fig. 8: Incremental object learning with different amounts of training data using the Washington dataset. *IC* is our approach and *Offline* is the offline baseline. (a) Accuracy vs training size. (b) Database size vs training size.



(a) (b)

Fig. 9: Incremental object learning with different amounts of training data from our MHRI dataset. *IC* is our approach and *Offline* is the offline trained baseline. (a) Accuracy vs training size. (b) Database size vs training size.

C. MHRI Dataset evaluation

Whereas the images in the Washington dataset are very clean, our MHRI dataset is better suited for an evaluation of a realistic scene and a natural human-robot interaction. We evaluated three descriptor options, as described in section V-A: RGB histogram (d_{RGB}), BOW histogram (d_{BoW}), and a combination of both (d_{ALL}).

Table V shows the accuracy of the baseline classifier for the three types of descriptor considered. The combination gives the best performance (75.50%). Notice that, as we advanced above, the performance of using only a BoW histogram descriptor is poor, as the low-resolution training views contains very few –if any– salient points.

Table VI shows the accuracy of our approach. The combination gives us the best score, it goes from 48% to 74%, very similar to top baseline performance. In Figure 9, we can see the progress of the performance for our incremental approach and the baseline. Our system's precision is a little lower than baseline, but, as we can see, the amount of data stored in our system is significantly less than baseline. Consequently, our system sacrifices a smidgen of precision for a considerable

d_{RGB}	74.74
d_{BoW}	31.48
d_{ALL}	**75.50**

TABLE V: Accuracy (%) for the offline trained baseline classifier using different descriptors and our MHRI dataset.

% train. data	20%	30%	40%	50%	60%	70%	80%	90%
d_{RGB}	47.5	57.7	65.0	68.7	69.9	70.1	72.2	72.2
d_{ALL}	48.2	57.6	64.7	68.8	70.6	71.1	72.8	73.3

TABLE VI: Accuracy (%) of our approach using different descriptors and the presented MHRI dataset. Columns represent the amount of data used for training.

amount space, often costly in robotic and embedded systems.

Figure 10 shows the time expenditure for processing a single image while training and testing. In training, the time is in the range between 1 and 25 milliseconds. There is fluctuations depending on the amount of data for each class and if reorganization is required. In testing, the times range from 6 to 16 milliseconds. This range is smaller than the training range, and also depends on the amount of data in each class.

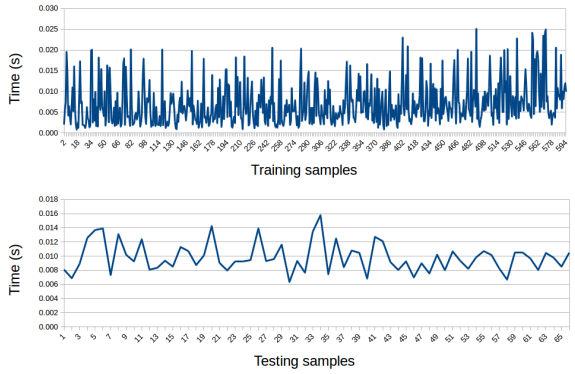


Fig. 10: Time used by our approach to process one image. The top plot is for training, the bottom one is for testing. The x-axis represents the number of samples trained or tested before processing this image. The y-axis displays the average time (seconds) for the 10 different executions from cross-validation.

As mentioned earlier, the fact that our object images, which come from the scenes with the user and the objects, influences the object segmentation. We designed an experimental setup to evaluate the magnitude of that influence. We separated our dataset into two groups. One contains all the *Show* interaction (in which the object can be occluded partially by the user holding it). The other group contains the remaining two interactions, *Point* and *Speak* (where the hand does not occlude the object). To perform the cross-validation we divided each group into blocks, 3 blocks for the occlusion group and 7 for the "clean" group. As in the experiments before, we enforced that at least two images per class are in each block and we use d_{ALL} as the descriptor.

The same experiment was run for both groups. We trained a model for each of the two groups, and performed three tests. For the first test, we trained each group's model with all the data from the opposing group and tested the model on each block. For the second test, we trained both group model on their own group data, and performed cross-validation within each group. For the third test, we trained both group model on their own group data, but performed cross-validation across all blocks of both groups. With this setup we intend to measure the influence of training on different interaction scenarios.

	Train&test from different domains		Train&test from the same domains		Train with all images	
	Mean	σ	**Mean**	σ	**Mean**	σ
Point,Speak	37.45	4.39	**89.30**	**4.61**	87.98	6.17
Show	26.77	4.05	51.25	7.81	48.83	9.20

TABLE VII: Accuracy mean and standard deviation for training/test sets of different/same domains.

In Table VII we can see that in the first test the accuracy is lower than previous experiments, around 26% with occlusion and 37% with the other. The second and the third test show that the occlusion group is more difficult to classify. The performance of the occlusion group is around 50% and the other group is around the 90%. Also we can see comparing the second and the third setup, that mixing the groups has little impact on the performance. This was to be expected for the clean images, but also illustrates that even with the occluded ones dividing the data can improve the precision.

D. Domain change

In this experiment we train the system with the Washington dataset images and use the test images of our dataset. Our aim is show experimentally the need for incremental learning, even for offline trained categories, due to the dataset bias.

As there is not a complete overlap between the objects in the Washington dataset and ours we cannot make a fair quantitative comparison in the same terms of the previous experiments. Figure 11 shows several qualitative examples. For the reduced test set that overlaps with the Washington dataset we observed that the accuracy was less than 20%.

VII. Conclusions

In this paper we have presented a framework for incremental and interactive learning of object models from multimodal robo-centric data. We also released an annotated multimodal dataset acquired in the context of the presented framework, but suitable for training and evaluation beyond this setting. We have presented and validated two recognition approaches for different stages of the whole interactive learning framework, both using the released dataset.

First, we have shown how even a simple task, classifying the type of user interaction from single frame global descriptors, can significantly benefit from multimodal data. The interaction recognition is a crucial step to facilitate future processing steps, such as automatic object and region of interest segmentation in images and videos. The contributions on this work are focused on the recognition modules of the human-robot interactive framework. In our incremental learning approach the object segmentation is done manually,

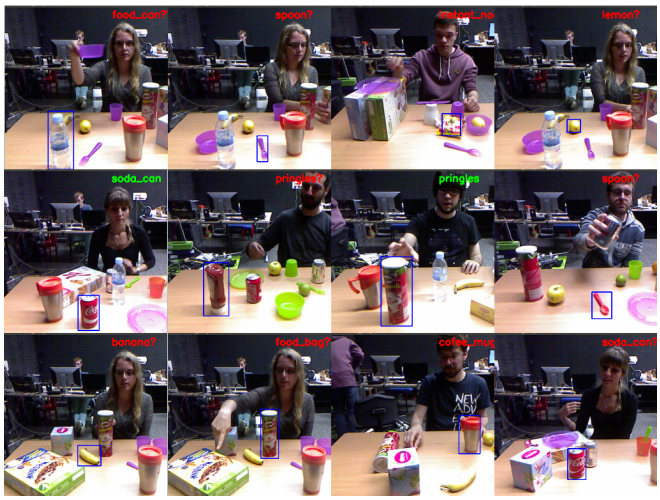


Fig. 11: Examples of our system trained with the Washington dataset and tested with our dataset. The output label is written in the image and the view used is shown in a blue rectangle.

therefore future lines of work will include the interaction recognition with automatic object segmentation.

Second, we have proposed an incremental approach to learn object models from the interaction with the user. Our experimental results show that our proposed system achieves results comparable to offline training, while operating on a much more limited amount of stored data. We evaluated our approach on both a standard public object recognition dataset, and in our multimodal dataset, which contains sample images of users interacting with objects and the robot. In addition to the interaction, our dataset presents other challenges like occlusions and low object resolution. A remarkable feature of the dataset is the synchronized recording of multimodal data, specifically with two *RGB-D* cameras, one high resolution *RGB* camera, and audio data. The incremental object learning approach presented here uses only visual data from a single camera. Future lines of work, taking advantage of the multimodal dataset released, will extend the presented approach by harnessing the higher modality.

References

[1] L. N. Abdullah and S. A. M. Noah. Integrating audio visual data for human action detection. In *Computer Graphics, Imaging and Visualisation, 2008. CGIV'08. Fifth International Conference on*, pages 242–246. IEEE, 2008.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[3] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer. Fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions on Robotics*, 24(5):1027–1037, 2008.

[4] J. Baraglia, M. Cakmak, Y. Nagai, R. Rao, and M. Asada. Initiative in robot assistance during collaborative task execution. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 67–74, March 2016.

[5] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. Sukhatme. Interactive perception: Leveraging action in perception and perception in action. *arXiv preprint arXiv:1604.03670*, 2016.

[6] R. Camoriano, G. Pasquale, C. Ciliberto, L. Natale, L. Rosasco, and G. Metta. Incremental object recognition in robotics with extension to new classes in constant time. *arXiv preprint arXiv:1605.05045*, 2016.

[7] A. Collet, D. Berenson, S. S. Srinivasa, and D. Ferguson. Object recognition and full pose registration from a single image for robotic manipulation. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 48–55, May 2009.

[8] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski. Self-supervised monocular road detection in desert terrain. In *Proceedings of Robotics: Science and Systems*, Philadelphia, USA, August 2006.

[9] J. Dumora, F. Geffard, C. Bidard, N. A. Aspragathos, and P. Fraisse. Robot assistance selection for large object manipulation with a human. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 1828–1833, Oct 2013.

[10] A. Gijsberts and G. Metta. Real-time model learning using incremental sparse spectrum gaussian process regression. *Neural Networks*, 41:59–69, 2013.

[11] W. Gong, J. Gonzàlez, J. M. R. S. Tavares, and F. X. Roca. *A New Image Dataset on Human Interactions*, pages 204–209. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[12] P. Iravani, P. Hall, D. Beale, C. Charron, and Y. Hicks. Visual object classification by robots, using on-line, self-supervised learning. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1092–1099. IEEE, 2011.

[13] B. S. A. S. Jaeyong Sung, Colin Ponce. Human activity detection from rgbd images. *AAAI workshop on Pattern, Activity and Intent Recognition (PAIR)*, 2011.

[14] J. Kenney, T. Buckley, and O. Brock. Interactive segmentation for manipulation in unstructured environments. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 1377–1382. IEEE, 2009.

[15] M. Krainin, B. Curless, and D. Fox. Autonomous generation of complete 3d object models using next best view manipulation planning. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5031–5037. IEEE, 2011.

[16] A. Kuznetsova, S. J. Hwang, B. Rosenhahn, and L. Sigal. Expanding object detector's horizon: incremental learning framework for object detection in videos. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28–36. IEEE, 2015.

[17] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset, May 2011.

[18] S. Lee, J. Lim, and I. H. Suh. Incremental learning from a single seed image for object detection. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 1905–1912. IEEE, 2015.

[19] G. Pasquale, C. Ciliberto, F. Odone, L. Rosasco, L. Natale, and I. dei Sistemi. Teaching icub to recognize objects using deep convolutional neural networks. *Proc. Work. Mach. Learning Interactive Syst*, pages 21–25, 2015.

[20] U. Reiser, C. P. Connette, J. Fischer, J. Kubacki, A. Bubeck, F. Weisshardt, T. Jacobs, C. Parlitz, M. Hägele, and A. Verl. Care-o-bot® 3-creating a product vision for service robot applications by integrating design and technology. In *IROS*, volume 9, pages 1992–1998, 2009.

[21] J. Rituerto, A. C. Murillo, and J. KoÅąecka. Label propagation in videos indoors with an incremental non-parametric model update. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2383–2389, Sept 2011.

[22] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. IEEE, 2011.

[23] J. Sinapov, C. Schenck, and A. Stoytchev. Learning relational object categories using behavioral exploration and multimodal perception. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5691–5698. IEEE, 2014.

[24] A. Singh, J. Sha, K. S. Narayan, T. Achim, and P. Abbeel. Bigbird: A large-scale 3d database of object instances. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 509–516. IEEE, 2014.

[25] A. Vatakis and K. Pastra. A multimodal dataset of spontaneous speech and movement production on object affordances. *Scientific Data*, 3:150078 EP –, Jan 2016. Data Descriptor.

[26] A. Yao, J. Gall, C. Leistner, and L. Van Gool. Interactive object detection. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3242–3249. IEEE, 2012.