Robotics and Autonomous Systems 62 (2014) 632–645

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Grounding semantic categories in behavioral interactions: Experiments with 100 objects

Jivko Sinapov *, Connor Schenck, Kerrick Staley, Vladimir Sukhoy, Alexander Stoytchev

Developmental Robotics Laboratory, Iowa State University, Ames, IA, United States

ARTICLE INFO

Article history:
Available online 9 November 2012

Keywords:
Semantic perception
Active and interactive perception
Category recognition
Learning and adaptive system
Behavior-based robotics
Developmental robotics

ABSTRACT

From an early stage in their development, human infants show a profound drive to explore the objects around them. Research in psychology has shown that this exploration is fundamental for learning the names of objects and object categories. To address this problem in robotics, this paper presents a behavior-grounded approach that enables a robot to recognize the semantic labels of objects using its own behavioral interaction with them. To test this method, our robot interacted with 100 different objects grouped according to 20 different object categories. The robot performed 10 different behaviors on them, while using three sensory modalities (vision, proprioception and audio) to detect any perceptual changes. The results show that the robot was able to use multiple sensorimotor contexts in order to recognize a large number of object categories. Furthermore, the category recognition model presented in this paper was able to identify sensorimotor contexts that can be used to detect specific categories. Most importantly, the robot's model was able to reduce exploration time by half by dynamically selecting which exploratory behavior should be applied next when classifying a novel object.

1. Introduction

Object categories are all around us—our homes and offices contain a vast multitude of objects that can be organized according to a diverse set of criteria ranging from form to function. A robot operating in human environments would undoubtedly have to assign category labels to novel objects because it is simply infeasible to preprogram it with knowledge about every individual object that it might encounter. For example, to clean a kitchen table, a robot has to recognize semantic object category labels such as silverware, dish, or trash before performing an appropriate action.

The ability to learn and utilize object category memberships is an important aspect of human intelligence and has been extensively studied in psychology [1]. A large number of experimental and observational studies have revealed that object category learning is also linked to our ability to acquire words [2,3]. Researchers have postulated that, with a few labeled examples, humans at various stages of development are able to identify common features that define category memberships as well as distinctive features that relate members and non-members of a target category [4,5]. Other lines of research have highlighted the importance of object exploration [6,7], which is important for learning object categories since many object properties cannot always be detected by passive observation [8,9].

Recently, several research groups have started to explore how robots can learn object category labels that can be generalized to novel objects [10–14]. Most studies have examined the problem exclusively in the visual domain or have used a relatively small number of objects and categories. To address these limitations, this paper proposes an approach to object categorization that enables a robot to acquire a large number of category labels from a large set of objects. This is achieved with the use of multiple behavioral interactions and multiple sensory modalities. To test our method, the robot in our experiment (see Fig. 1) explored 100 different objects classified into 20 distinct object categories using 10 different interactions (e.g., grasp, lift, tap, etc.) making this one of the largest object sets that a robot has physically interacted with.

Using features extracted from the visual, auditory, and proprioceptive sensory modalities, coupled with a machine learning classifier, the robot was able to achieve high recognition rates on a variety of household object categories (e.g., balls, cups, pop cans, etc.). The robot's model was also able to identify which sensory modalities and behaviors are best for recognizing each category label. In addition, the robot was able to actively select the exploratory behavior that it should try next when classifying an object, which resulted in faster convergence of the model's accuracy rates when compared to random behavior selection. Finally, the model was evaluated on whether it can detect if a novel object does not belong to any of the categories present in the robot's training set.

* Corresponding author.
E-mail addresses: jsinapov@iastate.edu (J. Sinapov), cschenck@iastate.edu (C. Schenck), kerrick@iastate.edu (K. Staley), sukhoy@iastate.edu (V. Sukhoy), alexs@iastate.edu (A. Stoytchev).

0921-8890/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2012.10.007

Fig. 1. The humanoid robot used in our experiments, along with the 100 objects that it explored.

2. Related work

Most object categorization methods in robotics fall into one of two broad categories: (1) unsupervised methods, in which objects are categorized using unsupervised machine learning algorithms (e.g., *k*-Means, Hierarchical Clustering, etc.) and (2) supervised methods, in which a labeled set of objects is used to train a recognition model that can label new data points. Several lines of research have demonstrated methods that enable robots to autonomously form internal object categories based on direct interaction with objects [15,11,16,17]. For example, Griffith et al. [11] showed how a robot can use the frequencies with which certain events occur in order to distinguish between container and non-container objects in an unsupervised manner. Dag et al. [16] and Sinapov and Stoytchev [18] have also shown that robots can categorize and relate objects based on the type of effects that they produce when an action is performed on them.

In contrast, the focus of this paper is on supervised methods for object categorization, which attempt to establish a direct mapping between the robot's object representation and human-provided semantic category labels. A wide variety of computer vision methods have been developed that attempt to solve the problem using visual image features coupled with machine learning classifiers [19–21]. Several such methods have been developed for use by robots, almost all exclusively working in the visual domain [22,23,12,24,14,25]. One advantage of visual object classifiers is that they can often be trained offline on large image datasets. Nevertheless, they cannot capture object properties that cannot always be perceived through vision alone (e.g., object compliance, object material, etc.). In other words, disembodied object category representations that are grounded solely in visual input cannot be used to capture object properties that require active interaction with an object. Thus, even the best visual classifier is guaranteed to fail on certain object classification tasks. For example, Lai et al. [26] report that using state-of-the-art RGB and depth features for classifying 300 objects into 51 categories results in 85.4% accuracy, which demonstrates that there is still a lot of information about object categories that cannot be captured using disembodied vision-based systems. Furthermore, it has been argued that embodied perception is not only desirable, but also required for achieving intelligent autonomous behavior by a robotic system [27]. Therefore, to address the limitation of disembodied systems, our robot grounded the semantic category labels of objects in its own sensorimotor experience with them, which is in stark contrast with approaches that rely purely on computer vision datasets.

The importance of non-visual sensory modalities for robotic object perception has been recognized by several lines of research, which have shown that robots can recognize objects using auditory [28–30], tactile [31,32], and proprioceptive [33,34] sensory modalities. For example, Natale et al. [33] showed that proprioceptive information obtained from the robot's hand when grasping an object can be used to successfully recognize the identity of the object. Similarly, Bergquist et al. [34] performed an experiment in which a robot was able to recognize a large number of objects using proprioceptive feedback from the robot's arm as it manipulated them. Other research has also shown that auditory features (e.g., sounds generated as the robot's end effector makes contact with an object) can also be useful for recognizing a previously explored object [28,29]. Most recently, a study by Sinapov et al. [35] demonstrated that a robot can achieve high object recognition rates when tested on a large set of 50 objects by integrating auditory and proprioceptive feedback detected over the course of exploring the objects. In contrast to this previous work, the study in this paper demonstrates that behavior-grounded object perception can also be used by a robot to both learn and recognize human-provided semantic category labels for novel objects.

Several studies have already demonstrated some ability of robots to assign category labels to objects based on interaction with them. For example, Takamuku et al. [36] demonstrated that a robot can classify 9 different objects as either a rigid object, a paper object, or a plastic bottle using auditory and joint angle data obtained when the robot shakes the objects. An experiment by Chitta et al. [37] has shown that tactile feedback produced during grasping can be useful for categorizing cans and bottles as either full or empty. In another study, Sinapov and Stoytchev [38] showed that by applying five different exploratory behaviors on 36 objects, a robot may learn to recognize their material type and whether they are full or empty, based on the auditory feedback produced by the objects.

In previous work, we proposed a graph-based learning method that allows a robot to estimate the category label of an object based on pairwise object similarity relations estimated from different couplings of five exploratory behaviors and two sensory modalities [13]. In that experiment, the robot was able to classify 25 objects according to object categories such as plastic bottles, objects with contents, pop cans, etc. The accuracy was substantially better than chance, despite the fact that visual feedback was not used.

To further improve category recognition rates, the study presented in this paper describes a method that scales to a much larger number of exploratory behaviors, sensory modalities, and objects than any previously published experiments in which robots have perceived objects by interacting with them. More specifically, in addition to doubling the number of objects, this paper also doubles the number of behaviors and more than triples the number of sensorimotor contexts as compared to our previous work [35] (which only focused on object recognition rather than category recognition). In addition, we also show that by using prior information in the form of confusion rates for all categories, the robot can actively select which behavior to apply next when classifying a novel object.

3. Experimental platform

3.1. Robot and sensors

The experiments were performed with the upper-torso humanoid robot shown in Fig. 1. The robot has as its actuators two 7-DOF Barrett Whole Arm Manipulators (WAMs), each with an attached 3-finger Barrett Hand. Each WAM has built-in sensors that measure joint angles and torques at 500 Hz. An Audio-Technica U853AW cardioid microphone mounted in the robot's head was used to capture auditory feedback at the standard 16-bit/44.1 kHz resolution and rate over a single channel. The robot's right eye

Fig. 2. The 100 objects explored by the robot, grouped in 20 object categories. From left to right and from top to bottom: (1) wicker baskets, (2) containers that vary by weight, (3) small stuffed animals, (4) large stuffed animals, (5) metal objects, (6) wooden blocks, (7) pasta boxes, (8) tin boxes (empty), (9) PVC pipes, (10) cups (vary by material), (11) pop cans, (12) plastic bottles, (13) food cans, (14) medicine pill bottles, (15) containers with different types of contents, (16) styrofoam cones, (17) foam noodles, (18) egg-coloring cups (vary only by color), (19) easter eggs (vary by material), and (20) balls.

(a Logitech webcam) captured 640 by 480 images that were used for visual feature extraction.

3.2. Objects

The robot explored 100 different household objects, which, to the best of our knowledge, is currently the largest number of objects explored by a robot over the course of a single experiment. The 100 objects were selected from 20 object categories, each containing 5 objects that vary along certain dimensions while remaining constant along others. For example, the 5 *PVC pipes* vary by width and weight, but have the same shape, color, and material type. Fig. 2 shows all objects and object categories that were used in the experiments.

3.3. Exploratory behaviors

The robot was equipped with 10 behaviors: *look*, *grasp*, *lift*, *hold*, *shake*, *drop*, *tap*, *poke*, *push*, and *press*. The *look* behavior consisted of simply taking an RGB snapshot of the object on the table (see Fig. 3). All other behaviors were encoded as trajectories in joint-space that were executed using Barrett's default PID controller (see Fig. 4). The only exceptions were the *grasp* and *tap* behaviors, which varied

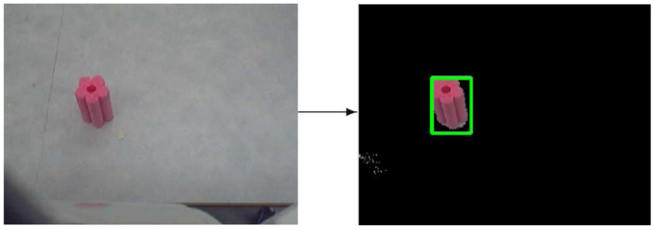

Fig. 3. Illustration of the visual object detection routine. The position of the bounding box around the object was used by the robot to apply the *grasp* and *tap* behaviors in the correct location (the remaining behaviors either assumed a fixed object position, or the robot was already holding the object). Features for visual object category recognition were extracted from the pixels corresponding to the object as described in Section 4.3.

depending on the visually detected initial position of the object.[1] It is worth mentioning that the proposed method for learning object categories is independent of how the behaviors are encoded.

3.4. Data collection

The robot interacted with the objects in a series of exploration trials. During each trial, an object was placed on the table by the experimenter and the robot performed all of its 10 exploratory behaviors on the object. The object was then switched with another object from the same category. This was repeated until the robot had explored each object from that category five times. If the objects within a given category could be placed in an order (e.g., by height or by weight), then they were explored in a sequence that is random with respect to the attribute by which they could be sorted. This process was repeated for all twenty categories. In the end, the robot had performed all 10 behaviors 5 times on each of the 100 objects, resulting in $10 \times 5 \times 100 = 5000$ behavior executions.

While performing each behavior, the robot recorded proprioceptive, auditory, and visual sensory feedback. The next section describes the feature extraction routines that were used to compute features from the recorded sensory input streams.

4. Feature extraction

4.1. Proprioceptive feature extraction

For each of the nine interactive behaviors shown in Fig. 4, proprioceptive features were extracted from the recorded joint torques from all 7 joints of the robot's left arm. The torques were recorded at 500 Hz. The joint-torque record from each interaction was represented as a $\mathbb{R}^{n\times 7}$ vector, where n is the number of temporal samples recorded for each of the 7 joints. Histogram features were extracted from each joint-torque record by discretizing the series of torque values for each joint into 10 temporal bins. This resulted in lower-dimensional datapoints $\mathbf{x} \in \mathbb{R}^{10\times 7}$, which were subsequently used for the tasks of training and applying the robot's category recognition model. Fig. 5 shows an example of this feature extraction process.

[1] Visual object detection was performed by estimating a background model of the table when there were no objects placed on it and using this model to fit a bounding box to the largest non-background connected component, which was assumed to be the object. Motor models for the *grasp* and *tap* behaviors were trained by repeatedly placing objects in various positions on the table and demonstrating initial and final joint angles for these behaviors by manually moving the robot's backdrivable arm. To synthesize these behaviors during object exploration, the robot used the three demonstrations closest to the current location of the object to compute average initial and final joint-space positions. The arm was then moved to these positions using the default PID controller.

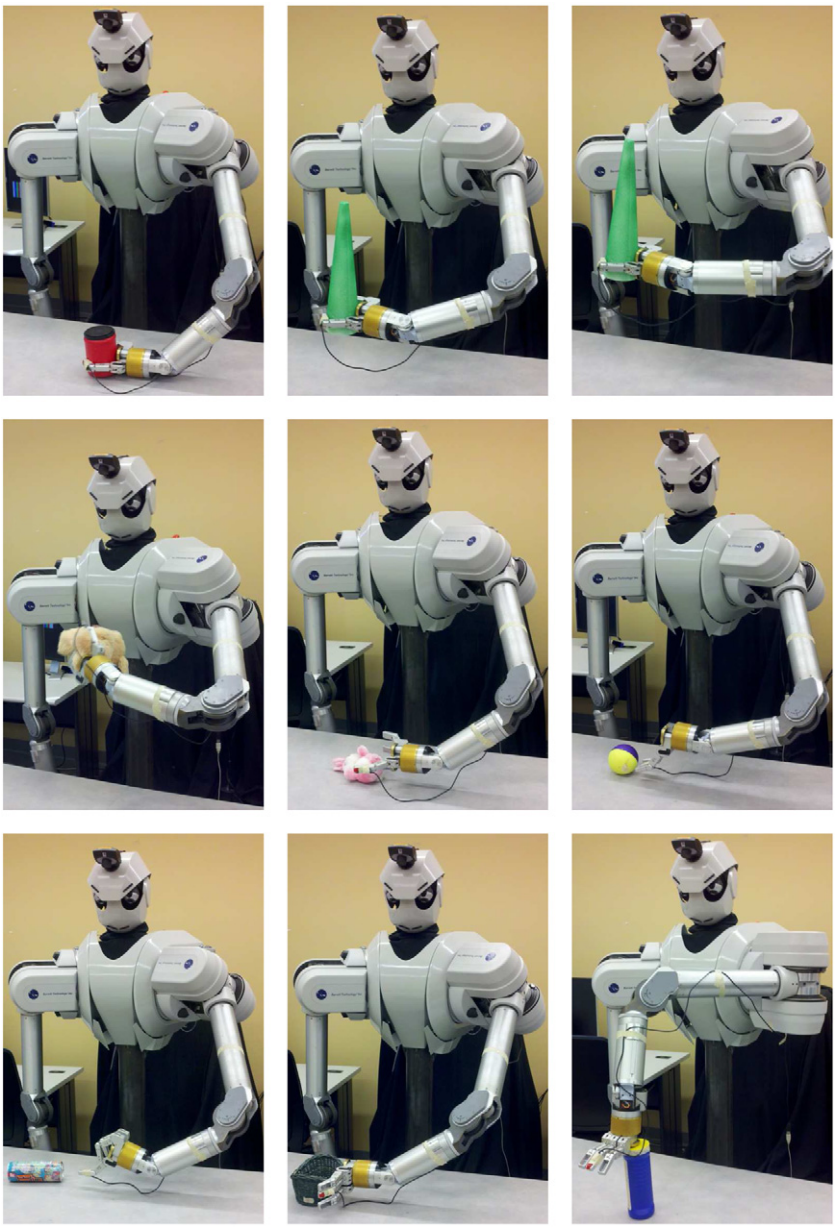

Fig. 4. The exploratory behaviors that the robot performed on all objects shown in Fig. 2. From top to bottom and from left to right: (1) grasp, (2) lift, (3) hold, (4) shake, (5) drop, (6) tap, (7) poke, (8) push, and (9) press. The *look* behavior is described in Fig. 3.

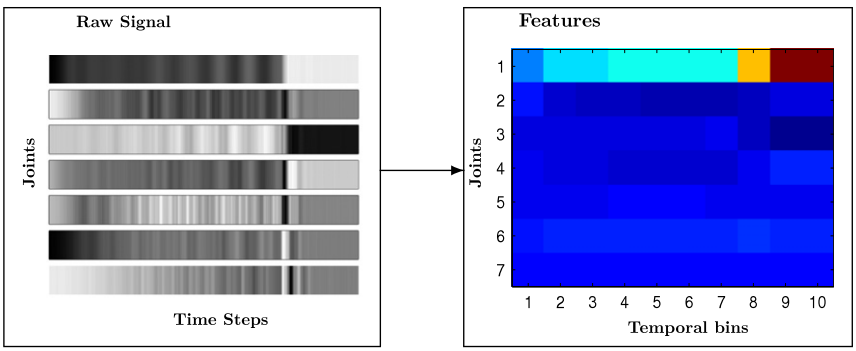

Fig. 5. Illustration of the proprioceptive feature extraction routine. The input signal is sampled during the execution of a behavior at 500 Hz and consists of the raw torque values for each of the robot's seven joints. Features are extracted by discretizing time (horizontal axis) into 10 temporal bins, resulting in a $7 \times 10 = 70$ dimensional feature vector.

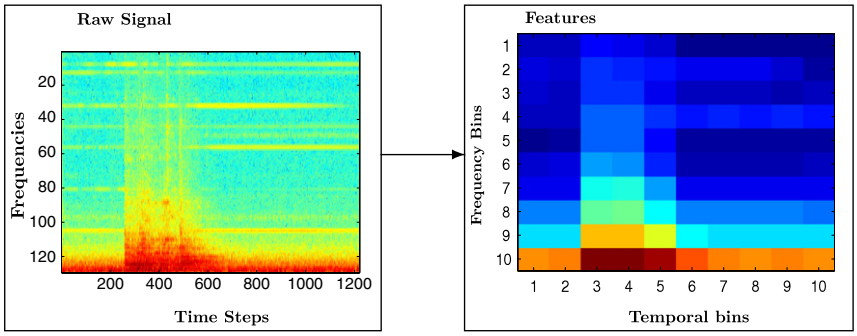

Fig. 6. Illustration of the auditory feature extraction procedure. The input consists of the discrete Fourier transform spectrogram of the audio wave recorded while a behavior is executed. The spectrogram encodes the intensity of 129 frequency bins and was calculated using a raised cosine window of 25.625 ms computed every 10.0 ms. To reduce the dimensionality of the signal both the time and the frequencies were discretized into 10 bins, resulting in a $10 \times 10 = 100$ dimensional feature vector.

4.2. Auditory feature extraction

Auditory features were extracted using the log-normalized Discrete Fourier Transform (DFT), which was computed for each detected sound using $2^7 + 1 = 129$ frequency bins. The SPHINX4 natural language processing library package was used to compute the DFT for each sound [39]. The DFT encoded the detected intensity for all 129 frequency bins over time, but it was highly-dimensional and thus could not be used directly as an input to the machine learning algorithm. Therefore, given a DFT matrix of a detected sound, a 2D histogram was computed by discretizing time into k_t bins and frequencies into k_f bins. The value for each bin in the histogram was set to the average of the values in the DFT matrix that fell into it. In all experiments, both k_t and k_f were set to 10. Thus, each sound was represented by a feature vector x, where $x \in \mathbb{R}^{10\times 10}$. Fig. 6 shows an example of this feature extraction routine.

4.3. Visual feature extraction

Three types of visual features were extracted from the output of the robot's RGB camera:

4.3.1. Color

During the execution of the *look* behavior, the recorded RGB image of the object was used to compute an $8 \times 8 \times 8$ (i.e., 512-dimensional) color histogram in RGB space with uniformly spaced bins. For each image, the object was segmented from the background to ensure that only pixels that correspond to the object are used in the computation of the histogram.

4.3.2. Optical flow

During the execution of all interactive behaviors, the stream of images captured by the robot's camera was used to extract optical flow features. To do so, the dense optical flow was first computed using the algorithm and MATLAB implementation proposed by Sun et al. [40]. More specifically, given an image from the raw video stream, for each pixel, the algorithm computes a two-dimensional real-valued vector (u, v) encoding the direction of motion (i.e., the vector's angle) as well as the magnitude of the motion (i.e., the vector's norm). The region of interest was set to include the whole image and captured motion produced both by the robot's arm and by the object. Fig. 7 illustrates this procedure. The optical flow data is very dense and cannot directly be used as an input to a machine learning algorithm. To overcome this, *weighted angular histogram* features were extracted from the sequence of optical flow images by binning the angles into 10 equally spaced bins. More specifically, the norms of all optical flow vectors with angles ranging from 0 to $2\pi/10$ are added to bin number 1, the norms of all vectors with angles in the range of $2\pi/10$–$2 \times 2\pi/10$ are added to bin number 2 and so forth.

4.3.3. SURF

The Speeded-Up Robust Features (SURF) proposed by [41] were computed for all images captured by the robot's camera during the execution of each of the 10 behaviors. Fig. 7 shows the detected SURF interest points for several images over the course of executing the *poke* behavior.

For the *look* behavior, the region of interest was set to the bounding box containing the segmented object from the background. For the remaining 9 behaviors, the SURF features were computed over a region of interest covering the entire table. Each SURF descriptor was represented as a 128-dimensional feature vector encoding the distribution of the first order Haar wavelet responses within the interest point neighborhood.

The detected SURF descriptors were quantized using the X-Means algorithm, an extension of k-Means that attempts to estimate the number of clusters using the Bayesian Information Criterion (see [42] for details). The quantization was learned using only 0.5% (or approximately 35,000) of the feature descriptors detected from all individual images captured by the robot's camera. The X-Means algorithm found 200 clusters that were interpreted as a dictionary of visual "words". Given a set of SURF descriptors detected over the course of executing a behavior on an object, a 200-dimensional feature vector was computed encoding a histogram of the SURF descriptors over the words in the dictionary.[2]

4.4. Hand proprioception feature extraction

The final configuration of the fingers at the end of the *grasp* behavior was also recorded. This resulted in a 3-dimensional feature vector, where each value indicates the end joint position for each of the three fingers of the Barrett Hand (BH-260). The final position of each finger was always in the range of 0 (fully open) to 20 000 (fully closed). The spread of the fingers (joint number 4) was held fixed during the execution of each grasp.

4.5. Summary

To summarize, the robot perceived the objects using 6 different types of features: (1) auditory, (2) proprioceptive (arm), (3) proprioceptive (hand), (4) color, (5) optical flow, and (6) SURF. The auditory, proprioceptive and optical flow features were extracted from the robot's sensorimotor data recorded while performing each of the 9 interactive behaviors on the objects. Color features, on the other hand, were extracted from the static images of the object taken by the robot's camera during the execution of

[2] Experiments were also conducted with larger visual word dictionaries, but no benefit to classification performance was observed.

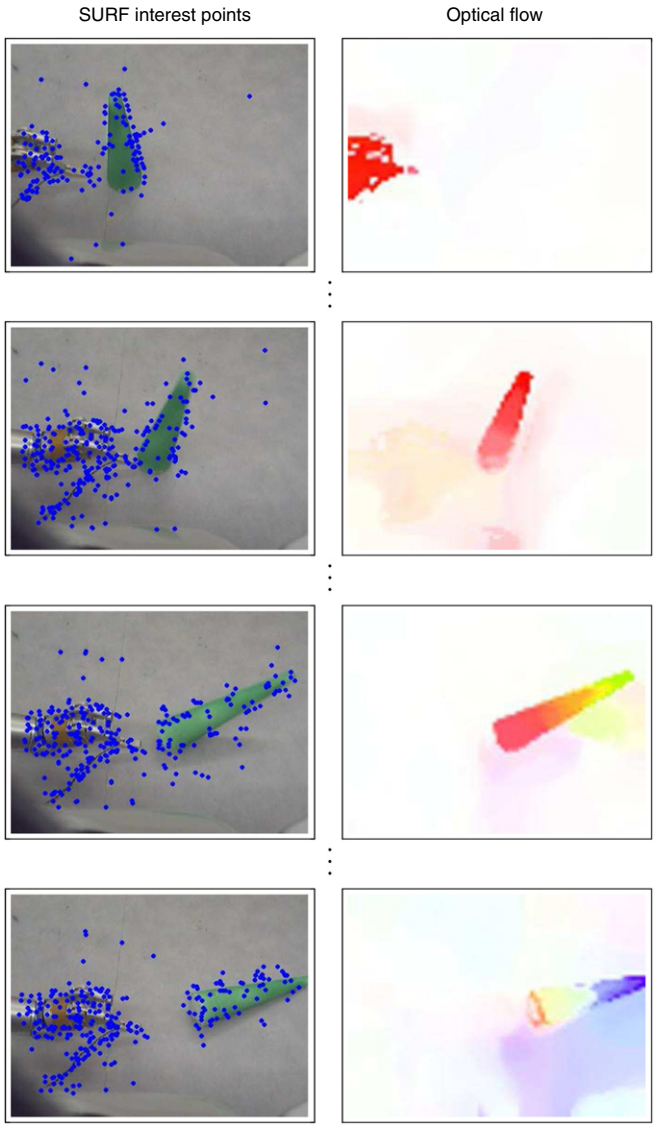

Fig. 7. Illustration of the SURF features and the optical flow detected through the robot's camera during the execution of the *poke* behavior on one of the objects from the *styrofoam cones* category. The left column shows the raw camera images with the detected SURF interest points, while the right column shows the corresponding optical flow images. For each pixel in the optical flow images, the hue encodes the angle of the optical flow vector (u, v) for that pixel, while the intensity encodes the vector's norm.

the *look* behavior. Finally, SURF features were extracted from both static images captured during the *look* behavior as well as the image sequences from the remaining 9 behaviors. The next section describes how these features are used for recognizing the category of an object.

5. Theoretical model

5.1. Notation

Let $\mathcal{B}$ be the set of exploratory behaviors and let $\mathcal{C}$ be the set of sensorimotor contexts such that each context $c \in \mathcal{C}$ refers to a combination of a behavior and a sensory modality (e.g., *drop-audio*, *look-color*, etc.). In our case, 9 behaviors (all except *look*) produced 3 types of feedback: auditory, optical flow, and proprioceptive feedback from the robot's arm. SURF features were extracted during all 10 behaviors. In addition, color features were extracted during the *look* behavior. Finally, the *grasp* behavior also produced proprioceptive feedback from the robot's hand. Thus, the total number of sensorimotor contexts in our experiments was $9 \times 3 + 10 + 1 + 1 = 39$. In other words, $|\mathcal{C}| = 39$.

Let $\mathcal{O}$ be the set of all 100 objects. During the data collection, the robot was repeatedly presented with an object $o \in \mathcal{O}$ and subsequently applied all of its exploratory behaviors on the object, which constituted one trial. Thus, during the ith exploration trial, the robot observed features x_i^c for each behavior–modality context c. The following subsections describe how these features can be used to solve the object category recognition task.

5.2. Problem formulation

Each object in our dataset was labeled as belonging to one of the 20 categories shown in Fig. 2. Let the function $label(o) \rightarrow y$ be a labeling function that outputs a label $y \in \mathcal{Y}$ given an object o, where $\mathcal{Y}$ is the full set of 20 category labels ($|\mathcal{Y}| = 20$). The task of the robot is to learn a category recognition model that outputs the correct category label y, given sensory feedback signals detected while interacting with object o using a set of behaviors $\mathcal{B}$.

5.3. Category recognition model

To solve this problem, for each sensorimotor context $c \in \mathcal{C}$, a category recognition model M_c is trained on input datapoints of the form $[x_i^c, y]$ where x_i^c is a feature vector detected in context c during trial i, while exploring an object with label y. The recognition model is tasked with estimating the category label probability for each class label, i.e., $\Pr(\hat{y} = y|x_i^c)$ for all labels $y \in \mathcal{Y}$. In this work, two different machine learning algorithms were evaluated: k-Nearest Neighbors (k-NN) and Support Vector Machine (SVM).

5.3.1. k-Nearest Neighbor

The first algorithm, k-Nearest Neighbors (k-NN), falls within the family of *lazy learning* or *memory-based learning* algorithms [43,44] and does not build an explicit model of the data. Instead, it simply stores all data points and their category labels and only uses them when the model is queried to label a test data point.

To label a test data point, k-NN finds its k closest neighbors in the training set. The Euclidean distance function (i.e., $L2$-norm) was used to calculate the distances between the test data point and the training samples when computing the set of k closest neighbors. The parameter k was heuristically set to 3. Probability estimates were computed by counting the category labels of the three neighbors. For example, if two of those neighbors have a class label "ball", then $\Pr(\hat{y} = \text{ball}) = 2/3$. All experiments were conducted using the implementation of k-NN included in the WEKA machine learning library [45].

5.3.2. Support vector machine

The second machine learning algorithm, Support Vector Machine (SVM), falls in the family of *discriminative* models [46]. Let $(\mathbf{x}_i, y_i)_{i=1,\ldots,l}$ be a set of labeled inputs, where $\mathbf{x}_i \in \mathbb{R}^n$ and $y_i \in \{-1, +1\}$ (i.e., a binary classification problem). The goal of the SVM algorithm is to learn a linear function $f(\mathbf{x}) = \langle \mathbf{x}, \mathbf{w} \rangle + b$, $\mathbf{w} \in \mathbb{R}^n$ and $b \in \mathbb{R}$, that can accurately classify test data points. To do this, the SVM algorithm solves a dual quadratic optimization problem, in which $\mathbf{w}$ and b are optimized so that the margin of separation between the two classes is maximized [46].

A good linear decision function $f(\mathbf{x})$ in the n-dimensional input space, however, does not always exist and therefore the labeled inputs are typically mapped into a (possibly) higher-dimensional feature space, e.g., $\mathbf{x}_i \rightarrow \Phi(\mathbf{x}_i)$, where a good linear decision function can be found. The mapping can be defined implicitly with a kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$ that replaces the

dot product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ in the dual quadratic optimization problem (see [46,47] for details). Intuitively, the kernel function can be interpreted as a measure of similarity between two data points.

In this work, several kernel functions were used. The first is the polynomial kernel function. Given two input feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j \in \mathbb{R}^n$, the polynomial kernel function is defined as:

$$K_{\text{poly}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^{\mathbf{T}} \mathbf{x}_j + 1.0)^p.$$

While the polynomial kernel function is one of the most commonly used ones in the literature, it is not appropriate for all types of data. Thus, two other kernel functions were also used, one designed to work on data points encoding a histogram [48] and another designed to handle data points that represent matrices rather than flat feature vectors [49]. Let $\mathbf{x}_i$ and $\mathbf{x}_j$ be two histograms such that $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{N}_0^n$, where $\mathbb{N}_0$ is the set of all non-negative integers. To handle histogram inputs, Chapelle et al. [48] propose the use of a non-Gaussian RBF kernel function:

$$K_{\text{hist}}(\mathbf{x}_i, \mathbf{x}_j) = e^{-\rho d_{a,b}(\mathbf{x}_i, \mathbf{x}_j)}$$

where

$$d_{a,b}(\mathbf{x}_i, \mathbf{x}_j) = \sum_k |x_{ik}^a - x_{jk}^a|^b.$$

If $a = 1$ and $b = 2$, this function corresponds to the commonly-used Gaussian RBF kernel. As Chapelle et al. [48] note, lowering b amounts to assuming that the data are generated by a mixture of distributions that are heavy-tailed when compared to the Gaussian distribution. Based on the experiments described in [48], in this work the parameters a and b were set to 1.0 and 0.5 respectively, while ρ was set to 0.1 (similar classification performance was observed as long as ρ was between 0.005 and 0.25). The K_{hist} kernel function was used by the SVMs trained on optical flow histogram features, and the SVMs trained on SURF histogram features, as well as the SVM trained to recognize the category of an object using its color histogram features.

Finally, since the auditory features correspond to a matrix (see Fig. 6), the auditory SVMs were trained using the trace kernel function designed to handle matrices [49]. Given two $n \times m$ matrices $\mathbf{X}_i$ and $\mathbf{X}_j$, the trace kernel function can be defined as:

$$K_{\text{trace}}(\mathbf{X}_i, \mathbf{X}_j) = \text{tr}(\mathbf{X}_i^{\mathbf{T}} \mathbf{X}_j)^p.$$

In summary, three different kernel functions were used in this work: K_{poly}, K_{hist}, and K_{trace}. The SVMs trained on optical flow angular histogram features and the SVM trained on color histogram features all used the K_{hist} kernel function. The SVMs trained on auditory features used the K_{trace} kernel functions. All other SVMs used the polynomial function, K_{poly}. The exponent p in K_{trace} and K_{poly} was set to 2.0.

To generalize the binary SVM classifier to the multi-class problem of category recognition, the pair-wise coupling method proposed by Hastie and Tibshirani [50] was applied in this work. Finally, to obtain probabilistic estimates from the SVM classifiers, logistic regression models were fitted to the outputs of the SVMs as described in [45]. The next subsection describes how the outputs from the context-specific category recognition classifiers were combined.

5.4. Combining model outputs

The outputs of several context-specific category recognition models can be combined in order to achieve better performance. The robot's experience with a given object o in multiple sensorimotor contexts during trial i can be represented by the set of features $\mathcal{X}_i = \{x_i^{c_1}, \ldots, x_i^{c_N}\}$, where each feature corresponds to the detected signal from a unique behavior–modality combination and N is the number of sensorimotor contexts ($N \leq |\mathcal{C}|$). The outputs of the individual models can be combined using the uniform combination rule:

$$\Pr(\hat{y} = y | \mathcal{X}_i) = \alpha \sum_{x_i^c \in \mathcal{X}_i} \Pr(\hat{y} = y | x_i^c)$$

where α is a normalization constant ensuring that the probabilities sum up to 1.0. By varying the number of elements in the input set $\mathcal{X}_i$, this formulation allows us to evaluate how the category recognition performance improves as the robot uses multiple sources of information.[3]

5.5. Active behavior selection

In practice, it would be highly desirable for a robot to minimize its object exploration time when classifying new objects. To address this challenge, the model in this work selected which behaviors to apply next based on prior information in the form of the confusion matrices associated with each behavior. More specifically, for a given behavior $b \in \mathcal{B}$, let $\mathbf{C}^b \in \mathbb{R}^{|\mathcal{Y}| \times |\mathcal{Y}|}$ be a confusion matrix such that each entry C_{ij}^b encodes how many times an object from category y_i was classified as belonging to category y_j.

Given a probabilistic estimate for an object's category, the confusion matrices associated with the robot's behaviors can be used to guide subsequent exploration. For example, suppose that after performing the *look* behavior, the robot's estimates for the object's category labels are $\Pr(\hat{y} = \text{“egg”}) = 0.6$, $\Pr(\hat{y} = \text{“ball”}) = 0.4$ and 0 for all others. Given this information, it may be possible to speed up exploration time if the next behavior that the robot chooses to apply is the one that confuses the “egg” and “ball” categories the least.

More specifically, for an exploratory behavior $b \in \mathcal{B}$, let $\Pr_b(\hat{y} = y_i | y = y_j)$ be the probability of mis-classifying an object from category y_j as an object from category y_i when applying behavior b. Thus, the degree of confusion between categories y_i and y_j for behavior b can be defined as:

$$C_{ij}^b = \frac{\Pr(\hat{y} = y_i | y = y_j) + \Pr(\hat{y} = y_j | y = y_i)}{2}.$$

The estimates for the confusion between categories are used by the robot to guide exploration as follows. Let $\hat{\mathbf{p}} \in \mathbb{R}^{|\mathcal{Y}|}$ be the robot's current probabilistic estimate for the object's category labels such that $\hat{p}_i$ is the probability that the object's category is y_i. Let $\mathcal{B}_r$ be the remaining set of behaviors to choose from (i.e., the behaviors not performed so far on the test object). In this setting, the next behavior to be applied is selected using the following procedure:

1. Compute the set $\mathcal{Y}_K \subset \mathcal{Y}$ such that it contains the K most likely object categories according to $\hat{\mathbf{p}}$.
2. Pick the next behavior b_{next} with an associated confusion matrix that is least likely to confuse the categories within the set $\mathcal{Y}_K$, i.e.,
$$b_{\text{next}} = \arg\min_{b \in \mathcal{B}_r} \sum_{y_i \in \mathcal{Y}_K} \sum_{y_j \in \mathcal{Y}_K / y_i} C_{ij}^b.$$
3. Update the estimate $\hat{\mathbf{p}}$ using the classifiers associated with the sensorimotor contexts of b_{next}.
4. Remove b_{next} from $\mathcal{B}_r$. If $|\mathcal{B}_r| \geq 1$, go back to step 1.

[3] Other combination rules that were explored include the product combination rule, a weighted combination rule, a majority vote rule, as well as a meta-learning approach in which the outputs of the individual classifiers were fed as input to a meta-learning classifier. The classification performance of these other rules was either nearly identical or slightly inferior to the rule used in this work. For a detailed review of different classifier combination schemes, see [51,52].

Rather than setting a static value for the threshold K, this value is determined on-line given the current estimate $\hat{\mathbf{p}}$ such that the likelihoods of the K most likely categories sum up to at least ω. For example, if there are only three categories, A, B, and C, with likelihood estimates 0.5, 0.4, and 0.1 respectively, and $\omega = 0.65$, then only the first two, A and B, will be included in $\mathcal{Y}_K$ since they are the two most likely categories and $0.5 + 0.4 > 0.65$. In our experiments, the value for the threshold ω was set to 0.65. The results remained similar provided that ω was between 0.5 and 0.8, with performance diminishing outside that range.

5.6. Detecting outlier categories

One limitation of the theoretical model presented so far is that it cannot handle objects that do not belong to any of the categories specified during training. This is an important problem because a robot operating in a human environment is guaranteed to encounter an object from a category that it has never been exposed to before. To handle such situations, this section describes a method that can enable the model to detect whether an object belongs to a known category or not.

The problem can be formulated as follows. Let o_{test} be a test object whose category label is unknown (it may be either from a known category or from an unfamiliar category). Let $\hat{y} \in \mathcal{Y}$ be the estimated category assigned to the object by the trained category recognition model. Finally, let the set $O_{\hat{y}} = \{o_1^{\hat{y}}, \ldots, o_n^{\hat{y}}\}$ contain the known objects from category $\hat{y}$. Given the object o_{test} and the set $O_{\hat{y}}$, the task is to detect whether or not o_{test} is from a novel category or not. In the machine learning literature, this problem is known as *outlier detection* (see [53] for a review). While there are many approaches to this problem, most typically assume a flat feature vector representation for the data, as well as large amounts of data points. Therefore, in this work, the method for detecting the presence of novel categories is based on an approach specifically designed to deal with a small number of objects that have been physically explored by a robot [54].

The original method proposed in [54] can be summarized as follows. Let $\mathbf{W} \in \mathbb{R}^{N \times N}$ be an affinity matrix encoding the similarity relations among a set $\mathcal{D}$ of N objects (i.e., data points). The outlier object is then selected as the object o_i that maximizes the following objective function:

$$q(\mathcal{D}, o_i) = \alpha_1 \sum_{j \in \mathcal{D}/o_i} \sum_{k \in \mathcal{D}/o_i} W_{jk} - \alpha_2 \sum_{j \in \mathcal{D}/o_i} W_{ij}.$$

The first term captures the pairwise similarity between the remaining objects in $\mathcal{D}$ (i.e., after i is removed from $\mathcal{D}$) while the second term captures the similarity between the selected object i and the remaining $|\mathcal{D}|-1$ objects in $\mathcal{D}$. The constants α_1 and α_2 are normalizing weights, which ensure that the function is not biased towards either one of the two terms. Thus, the weights were set to:

$$\alpha_1 = \frac{1}{(|\mathcal{D}| - 1) \times (|\mathcal{D}| - 1)}, \qquad \alpha_2 = \frac{1}{|\mathcal{D}| - 1}.$$

As reported in [54], given a set of physical objects explored by the robot, the proposed method is useful for detecting the object in the set that does not belong to the category. For example, given 3 pop cans and 1 hat, and a matrix encoding the similarity between the objects as measured by the sensorimotor features detected with the objects, the hat is selected as the odd object.

Given the object o_{test}, its estimated category label $\hat{y}$, the set of objects $O_{\hat{y}}$, and a similarity matrix $\mathbf{W}_c$ associated with sensorimotor context $c \in \mathcal{C}$, the method from [54] is adapted for outlier category detection using the following procedure:

- Let $o_{\text{odd}} = \arg\max_i q(O_{\hat{y}} \cup \{o_{\text{test}}\}, o_i)$.
- If $o_{\text{odd}} \neq o_{\text{test}}$, then classify the object o_{test} as belonging to the *familiar* category $\hat{y}$.
- If $o_{\text{odd}} = o_{\text{test}}$ and $q(O_{\hat{y}} \cup \{o_{\text{test}}\}, o_{\text{test}}) > \epsilon_{\hat{y}}^c$, then classify object o_{test} as belonging to a *novel* category. Else, classify o_{test} as an object from a known category, i.e., accept the estimated category label $\hat{y}$.

The threshold $\epsilon_{\hat{y}}^c$ is a parameter specific to the category $\hat{y}$ and context c, and is estimated from the available training data. This is done by repeatedly running the odd-one-out task on all groups of objects from the same category in the training set and recording the highest observed outlier score, $q_{\max}^{c,\hat{y}}$. Thus, $\epsilon_{\hat{y}}^c$ is set to $r \times q_{\max}^{c,\hat{y}}$, where $r \in \mathbb{R}$. For example, when $r = 1.0$, for an object to be considered an outlier, it has to have a higher odd-one-out score than the highest observed score for an object that belongs to the category.

Each entry in the matrices $\mathbf{W}_c$ is computed by estimating the expected similarity between the feature vectors detected with each pair of objects in context c. Given two feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ from the same context c, the similarity function that was used can be expressed as $e^{-\rho d_{L2}(\mathbf{x}_i, \mathbf{x}_j)}$ where d_{L2} is the $L2$ norm. The parameter ρ was heuristically set to 0.1, which is the same value as the one used in the definition of the SVM kernel function K_{hist} defined in Section 5.3.

The procedure for detecting the presence of an unfamiliar category takes as input just one similarity matrix $\mathbf{W}_c$, tied to a specific sensorimotor context. To use multiple sensorimotor contexts, the procedure is applied with several different matrices (one per sensorimotor context) and if more than half of the time the object o_{test} is detected as one from a novel category, then it is classified as such. In the experiments described in the next section, nine contexts were used for this task. This set of contexts was selected such that for each estimated category label $\hat{y}$, it contained the nine best contexts for recognizing category $\hat{y}$, as estimated by performing cross-validation on the training data.

5.7. Evaluation

5.7.1. Category recognition

The robot's category recognition models were evaluated using *object-based cross-validation* as follows. During each round of evaluation, the robot's context specific models were trained on data from 4 objects from each category (a total of 80 objects) and evaluated on data from the remaining 20 objects. This process was repeated five times, such that each object was included four times in the training set and once in the testing set. Since the robot explored each object over 5 trials, during the training stage each context-specific classifier was trained on $80 \times 5 = 400$ data points and evaluated on the remaining $20 \times 5 = 100$. For the purposes of this evaluation, outlier category detection was turned off to ensure that the classifiers are trained and tested using all available datapoints. Two metrics were used to quantify the category recognition performance. The first metric was accuracy, defined as:

$$\%\,\text{Accuracy} = \frac{\#\,\text{correct classifications}}{\#\,\text{total classifications}} \times 100.$$

The second metric was the *f-Measure*, which is defined as the harmonic mean between the precision and recall for a given category label. It can be computed as follows:

$$f\text{-Measure} = 2 \times \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}.$$

The f-Measure is always in the range of 0.0–1.0. For a given category, a high value of the f-Measure indicates that the category

Table 1
Category recognition accuracy (%) using the *look* behavior.

	Color histogram	SURF	All
k-NN	47.3	33.7	50.7
SVM	58.9	58.8	67.7

is easy to recognize, while a low value shows that the category is difficult to recognize.

In addition to evaluating the performance of the individual classifiers, the model's accuracy rates were also computed as the number of sensorimotor contexts available to the robot was varied from 1 to 39, and as the number of behaviors applied on the test object was varied from 1 to 10. For the latter case, both the random and the active behavior selection strategy were evaluated.

5.7.2. Outlier category detection

To evaluate the method for detecting the presence of novel categories, the initial set of categories was split into two groups of 10. The robot's category recognition models were subsequently trained with 4 out of 5 objects with the known category labels. In other words, the test set in this case contains data from 5 novel objects from each of the 10 novel categories as well as data from 1 novel object for each of the 10 familiar categories. The estimated category label for each object in the test set was computed using the trained category recognition model. Subsequently, the procedure described in Section 5.6 was used to decide whether to accept the category label or classify the object as one belonging to a category that was not present in the training set.

This test was repeated 20 times with different random seeds that determine how the set of categories is split into two groups. The results are reported in terms of *true positive rate*, i.e., the proportion of objects from novel categories that are classified as such, and *false positive rate*, i.e., the proportion of objects from familiar categories that are mistakenly classified as novel.

6. Results

6.1. Category recognition using a single behavior

The first experiment evaluated the performance of the robot's recognition models for each of the 39 possible sensorimotor contexts. Tables 1 and 2 show the accuracy rates for every viable combination of behavior and sensory modality.[4] The results show that nearly every sensorimotor context contains information useful for category recognition. For comparison, a model that randomly assigns an object category label is expected to achieve only 5.0% accuracy as the number of object categories is 20. On average, SVM performs substantially better than *k*-NN for most sensorimotor contexts.

As expected, certain behaviors work better with certain modalities. For example, the proprioceptive features detected during the *lift* behavior are more useful for object category recognition than the auditory features produced by the same object. One unexpected result is that auditory features produced by relatively silent behaviors such as *lift* and *hold* produce recognition accuracies better than chance. One possible explanation is that certain objects with contents inside of them (e.g., pasta boxes) still produce some auditory feedback that is indicative of the object's category. In addition, the sounds produced by the robot's motors while lifting and holding objects depend on the weight of the objects (i.e., heavier objects require larger torques). Another important result is that the SURF features detected over the course of manipulating the object are more useful for recognition than the features detected from the static *look* behavior. One possible explanation is that when performing a behavior, the object is observed from more than just one side and for a longer time frame, indicating that even if a robot uses only vision-based sensors to perceive objects, active interaction with them can still further improve the classification accuracy.

Table 2
Category recognition accuracy (%) using a single behavior.

	Behavior	Audio	Proprioception	Optical flow	SURF	All
k-NN	Grasp	30.9	38.9	13.6	48.3	64.0
	Lift	34.1	37.1	5.0	54.3	62.4
	Hold	20.4	24.5	5.0	39.5	43.6
	Shake	42.7	39.1	25.0	69.3	71.2
	Drop	45.7	18.8	16.0	40.5	59.0
	Tap	51.9	29.1	20.4	61.9	72.2
	Push	64.2	58.6	22.8	65.0	84.8
	Poke	48.5	53.1	18.8	57.7	76.0
	Press	46.7	66.1	24.0	59.7	69.6
SVM	Grasp	45.7	38.7	12.2	57.1	65.2
	Lift	48.1	63.7	5.0	65.9	79.0
	Hold	30.2	43.9	5.0	58.1	67.0
	Shake	49.3	57.7	32.8	75.6	76.8
	Drop	47.9	34.9	17.2	57.9	71.0
	Tap	63.3	50.7	26.0	77.3	82.4
	Push	72.8	69.6	26.4	76.8	88.8
	Poke	65.9	63.9	17.8	74.7	85.4
	Press	62.7	69.7	32.4	69.7	77.4

To visualize the errors made by the robot's collection of recognition models, the 39 confusion matrices associated with the 39 sensorimotor contexts were summed up, producing the matrix $\mathbf{M} \in \mathbb{Z}^{20\times 20}$ in which each entry M_{ij} encodes how many times category i was confused with category j. A second, symmetric matrix $\mathbf{M}^{\text{sym}}$ was then computed such that $M_{ij}^{\text{sym}} = M_{ij} + M_{ji}$. The matrix $\mathbf{M}^{\text{sym}}$ was then used to produce a taxonomy of the categories by recursively applying the normalized-cut algorithm [55]. The result is shown in Fig. 8. Categories that are likely to be confused by at least some of the classifiers in the ensemble are close within the taxonomy while categories that are easy to distinguish are further apart. While the taxonomy is not expected to match how a human would organize the categories, it still shows how perceptually similar they are from the robot's point of view.

6.2. Category recognition from multiple sensorimotor contexts

The next experiment evaluated whether the robot's category recognition performance could be improved by combining the outputs of individual recognition models trained on data from specific behavior–modality combinations. As before, the models were trained with known labels for 4 out of the 5 objects in each category and evaluated on the remaining set. In this case, however, the evaluation was performed by varying the number of sensorimotor contexts that were used for classifying a novel object from 1 to 39 (see Section 5.4 for details on how the outputs from multiple context-specific recognition models are combined). Due to the large number of tests that need to be performed for this experiment, only *k*-NN was evaluated with a variable number of sensorimotor contexts available to the robot.

Fig. 9 shows the categorization performance for each of the 20 object categories as the number of contexts is varied from 1 to 39. As the robot is allowed to experience objects in more sensorimotor contexts its ability to classify them into categories increases. Most object categories (14 out of 20) can be recognized almost perfectly (i.e., f-Measure greater than 0.9) if all sources of information are

[4] For the *grasp* behavior and the proprioceptive sensory modality, the outputs of the *arm* and *hand* proprioceptive recognition models were combined and the resulting accuracy is reported. Individually, the arm proprioceptive model achieved an accuracy of 36.27%, while the hand proprioceptive model achieved 21.84% when using the *k*-NN algorithms. With SVM, the rates were 36.7% and 21.5%, respectively.

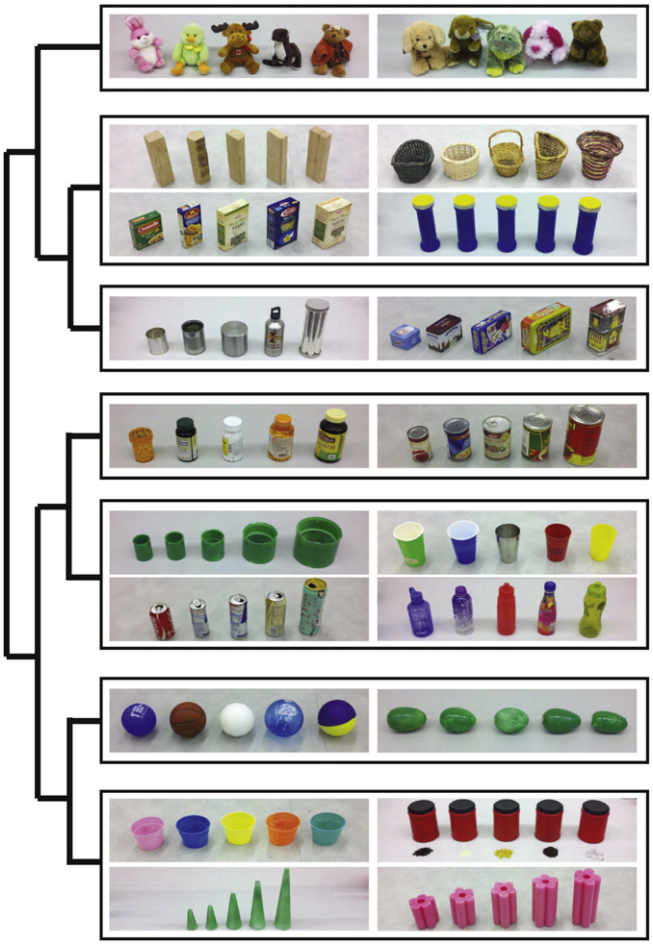

Fig. 8. A hierarchical clustering of the 20 categories based on the confusion matrix encoding how often each pair of categories is confused by the robot's context-specific category recognition models.

used. When all 39 sensorimotor contexts are used, k-NN achieved 94.6% category recognition accuracy. The SVM algorithm was also evaluated when using all 39 contexts, resulting in 97% accuracy.

Table 3 shows the specific precision and recall rates for all 20 categories when using all 39 contexts. The object category that was most difficult to recognize was the *metal objects* category, for which the f-Measure was only 0.57. Objects from this category were most often mis-classified as belonging to the *tin boxes* category, which was likely due to the fact that both of these categories consisted of objects that were made of metal. This illustrates that for a large set of objects it may be difficult to specify perfectly disjoint category assignments. In future work, we plan to address this by devising a category recognition method that can handle objects that may belong to multiple categories.

6.3. Identifying task-relevant sensorimotor contexts

The previous experiment showed that the robot can improve its category recognition performance by using information from all available sensorimotor contexts as opposed to just one. Nevertheless, this may not result in optimal recognition rates as certain contexts may produce features that are irrelevant for a given object category, thus making the learning task more difficult. To address this issue, in the next set of experiments the robot was tasked with estimating the most useful sensorimotor contexts for recognizing a given category.

To do so, during the training stage, the model performed internal cross-validation on the training data for each possible context–category combination, and the resulting f-Measure was recorded. At test time, for each category, the three contexts with

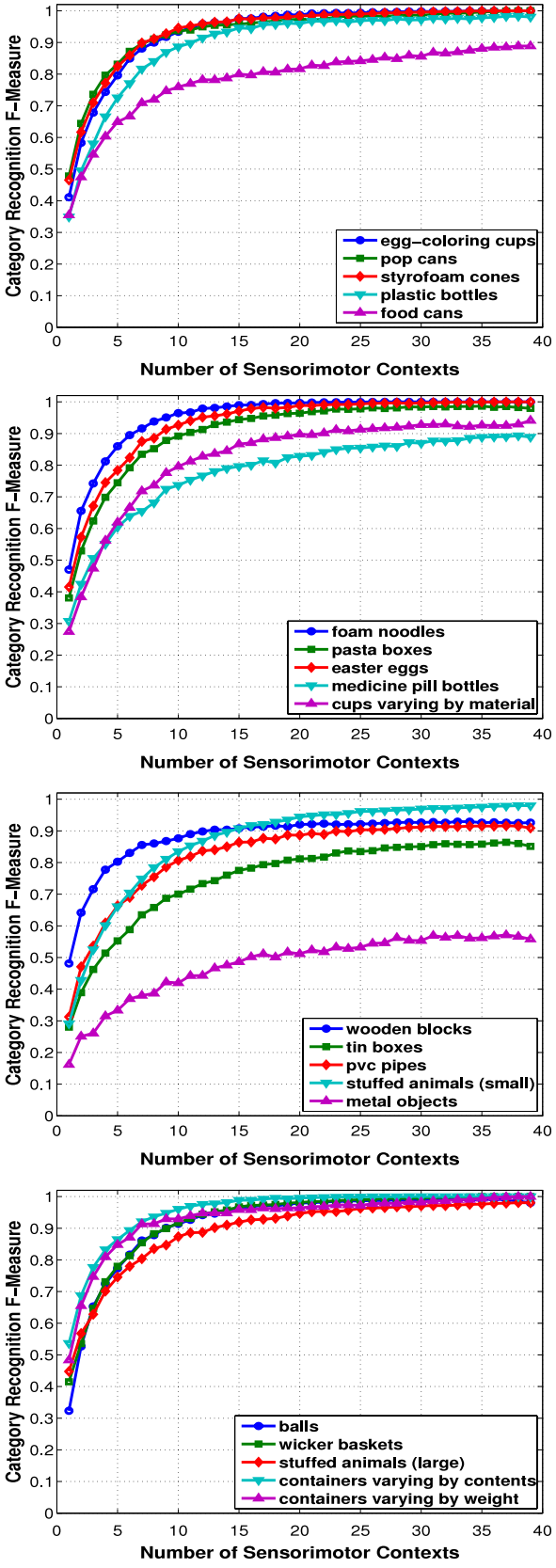

Fig. 9. Category recognition rates as a function of the number of sensorimotor contexts from which features are extracted. The results of this experiment show that the f-Measure increases dramatically as the robot experiences the objects using more behaviors and more sensory modalities.

Table 3
Precision and recall rates for all 20 categories using all sensorimotor contexts.

Category	*k*-NN		SVM	
	Precision	Recall	Precision	Recall
Wicker baskets	1.0	1.0	1.0	1.0
Containers (vary by weight)	0.93	1.0	1.0	1.0
Small stuffed animals	1.0	0.96	1.0	0.96
Large stuffed animals	0.96	1.0	0.96	1.0
Metal objects	0.67	0.48	0.64	0.76
Wooden blocks	0.86	1.0	0.96	1.0
Pasta boxes	1.0	0.96	1.0	1.0
Tin boxes	0.91	0.8	0.77	0.96
PVC pipes	0.82	1.0	1.0	0.96
Cups	0.89	0.96	1.0	1.0
Pop cans	1.0	1.0	1.0	1.0
Plastic bottles	0.96	1.0	1.0	1.0
Food cans	1.0	0.8	0.96	0.92
Medicine pill bottles	0.83	0.96	0.89	1.0
Containers (vary by contents)	1.0	1.0	1.0	1.0
Styrofoam cones	1.0	1.0	1.0	1.0
Foam noodles	1.0	1.0	1.0	1.0
Egg-coloring cups	1.0	1.0	1.0	1.0
Easter eggs	1.0	1.0	1.0	1.0
Balls	1.0	1.0	1.0	1.0

the highest f-Measures were used for detecting whether a novel object was a member of that category or not. Note that the set of 5 best contexts for each category is not necessarily the best *combination* of five contexts.

Fig. 10 shows histograms of the category recognition rates (f-Measure) for three different conditions: (1) using the 5 best sensorimotor contexts (top); (2) using 5 random sensorimotor contexts (middle); and (3) using all 39 sensorimotor contexts (bottom). The results show that the robot is able to identify a group of five task-relevant sensorimotor contexts that can be used to detect specific categories with performance comparable to that of using all 39 sensorimotor contexts. In other words, for each category, there exists a set of 5 contexts for which the performance is close to that achieved when using all sensorimotor contexts. Thus, if the robot is tasked with finding objects from a specific category, it could do this more efficiently by only applying the behaviors that are included in these 5 sensorimotor contexts.

It is important to note that the best sensorimotor features will be different for different categories. For example, the best sensorimotor context for the *blue containers* category was the *look-color* behavior–modality combination since the objects in that category vary by weight but are identical in color. The same combination, however, was not very useful for categories with objects that vary by color. The *egg coloring cups* category, for example, was easiest to recognize in the *press-proprioception* sensorimotor context since that context implicitly captures some of the objects' geometry and compliance (the objects in that category were identical in shape, height, and material type). For certain categories, auditory feedback was most useful for recognition. For example, the single best context for the *wooden blocks* category was *tap-audio* since wooden objects produce a distinct sound when tapped by the robot's fingers.

6.4. Active behavior selection

In practice, it may also be useful to know how many behaviors need to be performed to achieve a desired accuracy rate. To obtain this result, the number of behaviors performed at test time is varied from 2 to 10 under two different conditions: random behavior selection and active behavior selection (see Section 5.5). When evaluating the performance for active behavior selection, the first behavior is always chosen at random.

Fig. 11 shows the result of this test in which both models converge to 94.6% when using all 10 behaviors. However, when

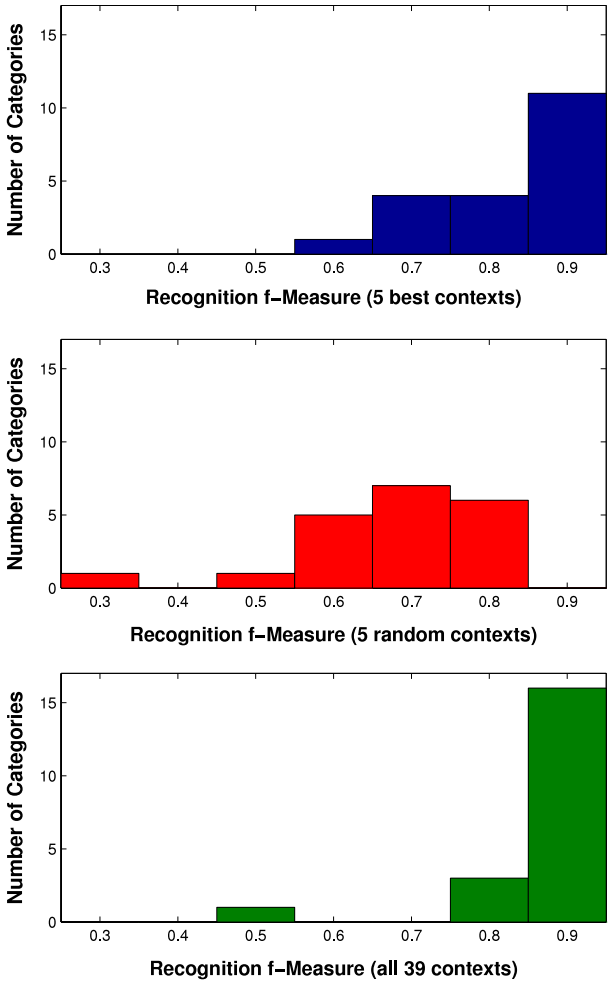

Fig. 10. Histograms of individual f-Measures per object category under three different conditions: (top) when using the 5 best contexts for each category; (middle) when using 5 random contexts; and (bottom) when using all 39 sensorimotor contexts. The results show that by identifying which 5 sensorimotor contexts work best for a given category the robot's model can improve its recognition when compared to any random combination of the same number of contexts.

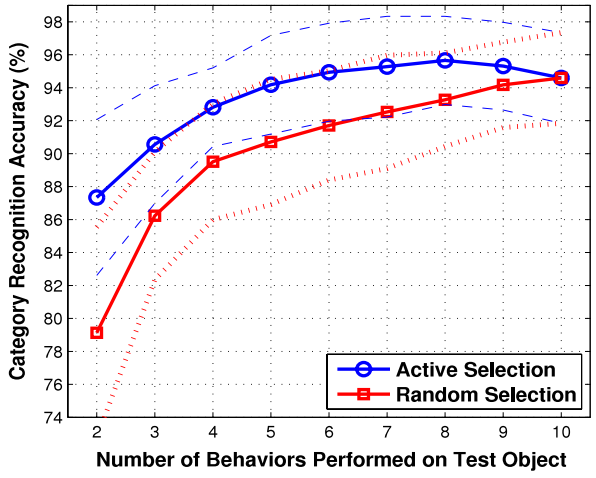

Fig. 11. Category recognition rates with k-NN classifier as a function of the number of behaviors applied on the test object under two different conditions: random behavior selection and active behavior selection (see Section 5.5). For each condition, the evaluation was performed using 5 different train–test splits. For each of the five splits, the evaluation was performed using each of the 10 behaviors as an initial state. Thus, the means and the standard deviations were computed from samples of size 50.

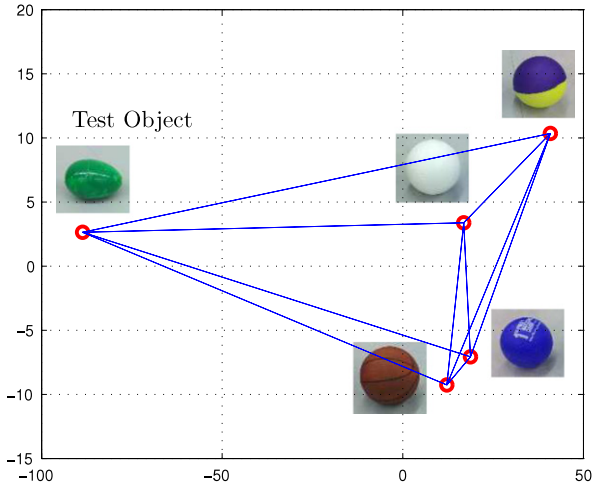

Fig. 12. A sample case of outlier category detection. In this example, the category *easter eggs* is not present in the robot's training set. Initially, the test object (one of the eggs) is classified as belonging to the *balls* object category by the robot's recognition model. The graph represents a 2-dimensional ISOMAP embedding of a context-specific distance matrix between the 5 objects, i.e., the four known balls and the egg, which is the test object. The sensorimotor context in this example was *press-proprioception*. The distance matrix is converted to a similarity matrix and the procedure outlined in Section 5.6 is applied to detect whether the test object should indeed be classified as a ball, or whether it should be considered as one belonging to a novel category. In this case, the method correctly detects that the egg should be considered as belonging to a category not present in the robot's training set.

randomly selecting the next behavior, the performance of the model crosses the 94% threshold after the 8th behavior. On the other hand, the active behavior selection strategy converges to the same rate after only the 4th behavior, i.e., the exploration time during testing is reduced by half. An interesting observation is that the active behavior selection strategy can achieve higher performance with slightly less than all 10 exploratory behaviors. A possible explanation for this is that under the active strategy, the last one or two behaviors that remain are the behaviors that are least accurate for the category of the test object, and thus their output acts as noise in the final combination.

6.5. Detecting outlier categories

In the last set of experiments, the robot's model was tasked with inferring whether a novel object belongs to a category that is not present in the robot's training set of categories. Fig. 12 shows a sample case in which the category of the test object (*easter eggs*) is not actually present in the robot's training set. Initially, the category recognition model incorrectly classified the test object as a ball, most likely because the egg has many similar properties as the balls (e.g., shape, size, etc.). Next, the procedure for detecting the odd-one-out object (described in Section 5.6) was applied, and in this case, the egg was selected as the outlier. As a result, the estimated category label (*balls*) for the test object was rejected and instead, the object was classified as belonging to a novel category. The figure shows an ISOMAP embedding [56] of the matrix encoding the pair-wise distances between all five objects, as computed in the *press-proprioception* sensorimotor context. As can be seen from the figure, the four balls form a tight cluster in this context and the egg is easily identified as the odd-one-out.

Fig. 13 shows the results after the entire evaluation, for different values of the constant r, which determines the necessary threshold that must be exceeded before the object is classified as belonging to an unfamiliar category. The results are reported in terms of true positive rate (the proportion of objects from novel categories classified as such) and false positive rate (the proportion of objects from familiar categories that are mistakenly classified as novel ones). When r is in the range of 1.5–2.0, most objects from novel

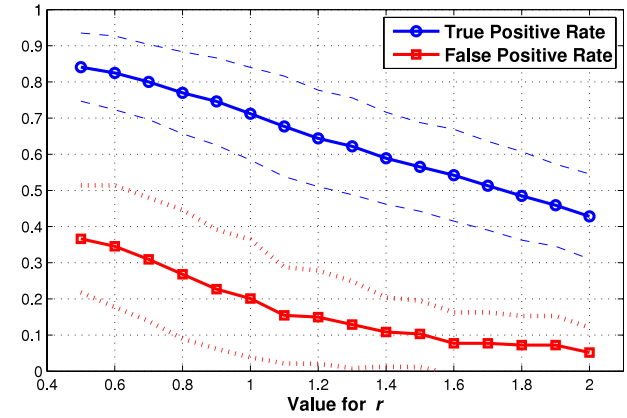

Fig. 13. Evaluation of the robot's model for detecting the presence of unknown categories. The results are reported in terms of true positive rate (i.e., the proportion of objects from novel categories classified as such), and false positive rate (i.e., the proportion of objects from familiar categories that are mistakenly classified as novel ones). The model is evaluated for different values of the constant r, which determines the threshold that needs to be exceeded for an object to be classified as belonging to an outlier category.

categories can be detected as such, while only a small number of objects from familiar categories are falsely classified as novel.

A large portion of the mistakes made by the model involved the *metal objects* category. For example, when the *pop can* category was not present in the training set, objects from it were classified as belonging to the *metal objects* category. Since a pop can is made of metal, the odd-one-out method was not able to clearly separate it from the known metal objects. In other words, many of the mistakes reflect the fact that specifying a perfectly disjoint object categorization for a large set of objects is nearly impossible.

7. Conclusion and future work

The ability to classify objects into categories is a pre-requisite for intelligent manipulation in human environments. To solve a wide variety of household tasks – from sorting objects on a table, to cleaning a kitchen, to taking out the trash – a robot must be able to recognize the semantic category labels of novel objects in its environment. This paper addressed the problem of object category recognition by presenting an approach that enables a robot to acquire a rich sensorimotor experience with objects and subsequently use visual, auditory, and proprioceptive features to label them. Using simple sensorimotor features coupled with the k-NN and SVM classifiers, the category recognition model was able to scale up to a large number of objects with a diverse set of category labels. Our method was tested using a large-scale experiment in which the robot repeatedly interacted with 100 different objects from 20 object categories using 10 different behaviors (e.g., looking at the object, grasping it, shaking it, tapping it, etc.). The high recognition rates achieved by the robot (e.g., 97% using SVM) show that perceiving objects using a diverse set of behaviors and sensory modalities is crucial for scaling up object category recognition to a large number of objects and object categories. The model was also able to identify task-relevant sensorimotor contexts for a given categorization task, which allow a robot to learn what specific behaviors and sensory modalities are best for recognizing a specific category label in a novel object. Most importantly, by actively selecting which behavior to apply next, the model was able to reduce by half the exploration time required for classifying a new object. Finally, the robot's model was extended to detect if the test object does not belong to any of the known categories.

There are several direct lines for future work that can further improve the robot's categorization skills. First, a limitation of the current system is that many of the features used to train the

classifiers are not invariant with respect to many aspects of the environment that were fixed in the lab setting (e.g., background audio noise, etc.). While much work in the computer vision literature has focused on identifying and computing features that are invariant with respect to scale, orientation, and illumination, it is still an open research question how to do the same for other sensory channels such as audio and proprioception. In addition, some level of invariance to changes in the environment can also be attained by employing machine learning methods that assume that the input data is sampled from a non-stationary distribution (see [57] for a review).

Second, it would be highly desirable to relax the assumption that all objects in the robot's training set have corresponding category labels since it may be infeasible to provide such category assignments for all objects that a robot interacts with. This problem can be addressed by using semi-supervised learning methods [58,59] that can make use of both labeled and unlabeled data. Furthermore, since real world objects typically belong to more than one category, it may be desirable to employ a multi-label classification paradigm (see [60] for a review). This can be achieved by either transforming the multi-label problem into a set of standard classification tasks (e.g., the method proposed in [61]) or by employing machine learning algorithms that are directly adapted to the multi-label data representation (e.g., the multi-label AdaBoost method proposed in [62]).

Finally, while in this paper the robot was able to perform all of its behaviors on all 100 objects, this may not be feasible if the number of objects is scaled up to 1000 or more. Instead of exhaustively exploring the objects, a robot dealing with such a large number of objects would need to apply behaviors in a way that minimizes exploration time but maximizes the relevant information extracted from the objects. One way to address this problem is to apply models of intrinsic curiosity and motivation [63] to behavior-grounded object exploration. Along those lines, advanced methods for classifier selection (e.g., [64]) could also be explored to further reduce the number of interactions required to correctly classify an object.

References

[1] F. Ashby, W. Maddox, Human category learning, Psychology 56 (1) (2005) 149.
[2] A. Fulkerson, S. Waxman, Words (but not tones) facilitate object categorization: evidence from 6- and 12-month-olds, Cognition 105 (1) (2007) 218–228.
[3] K. Plunkett, J. Hu, L. Cohen, Labels can override perceptual categories in early infancy, Cognition 106 (2) (2008) 665–681.
[4] R. Hammer, G. Diesendruck, D. Weinshall, S. Hochstein, The development of category learning strategies: what makes the difference? Cognition 112 (1) (2009) 105–119.
[5] R. Hammer, A. Brechmann, F. Ohl, D. Weinshall, S. Hochstein, Differential category learning processes: the neural basis of comparison-based learning and induction, NeuroImage 52 (2) (2010) 699–709.
[6] E.J. Gibson, Exploratory behavior in the development of perceiving, acting, and the acquiring of knowledge, Annual Review of Psychology 39 (1988) 1–41.
[7] T.G. Power, Play and Exploration in Children and Animals, Lawrence Erlbaum Associates, Publishers, Mahwah, NJ, 2000.
[8] M. Ernst, H. Bulthof, Merging the senses into a robust percept, Trends in Cognitive Sciences 8 (4) (2004) 162–169.
[9] D. Lynott, L. Connell, Modality exclusivity norms for 423 object properties, Behavior Research Methods 41 (2) (2009) 558–564.
[10] L. Lopes, A. Chauhan, Scaling up category learning for language acquisition in human–robot interaction, in: Proceedings of the Symposium on Language and Robots, 2007, pp. 83–92.
[11] S. Griffith, J. Sinapov, M. Miller, A. Stoytchev, Toward interactive learning of object categories by a robot: a case study with container and non-container objects, in: Proceedings of the 8th IEEE International Conference on Development and Learning, 2009.
[12] Z. Marton, R. Rusu, D. Jain, U. Klank, M. Beetz, Probabilistic categorization of kitchen objects in table settings with a composite sensor, in: Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, pp. 4777–4784.
[13] J. Sinapov, A. Stoytchev, Object category recognition by a humanoid robot using behavior-grounded relational learning, in: Proceedings of the 2011 IEEE International Conference on Robotics and Automation, ICRA, 2011, pp. 184–190.
[14] A. Leonardis, S. Fidler, Learning hierarchical representations of object categories for robot vision, Robotics Research (2011) 99–110.
[15] T. Nakamura, T. Nagai, N. Iwahashi, Multimodal object categorization by a robot, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007, pp. 2415–2420.
[16] N. Dag, I. Atil, S. Kalkan, E. Sahin, Learning affordances for categorizing objects and their properties, in: Proc. of the IEEE International Conference on Pattern Recognition, 2010, pp. 3089–3092.
[17] J. Sun, J. Moore, A. Bobick, J. Rehg, Learning visual object categories for robot affordance prediction, The International Journal of Robotics Research 29 (2–3) (2010) 174.
[18] J. Sinapov, A. Stoytchev, Detecting the functional similarities between tools using a hierarchical representation of outcomes, in: Proceedings of the 7th IEEE International Conference on Development and Learning, 2008, pp. 91–96.
[19] R. Fergus, P. Perona, A. Zisserman, A visual category filter for google images, Lecture notes in computer science (2004) 242–256.
[20] J. Ponce, Toward Category-Level Object Recognition, vol. 4170, Springer-Verlag New York Inc., 2006.
[21] A. Opelt, A. Pinz, M. Fussenegger, P. Auer, Generic object recognition with boosting, IEEE Transactions on Pattern Analysis and Machine Intelligence (2006) 416–431.
[22] L. Lopes, A. Chauhan, Scaling up category learning for language acquisition in human–robot interaction, in: Proceedings of the Symposium on Language and Robots, 2007, pp. 83–92.
[23] K. Lai, D. Fox, 3D laser scan classification using web data and domain adaptation, in: Proc. of Robotics: Science and Systems, RSS, 2009.
[24] W. Wohlkinger, M. Vincze, 3D object classification for mobile robots in home-environments using web-data, in: IEEE 19th International Workshop on Robotics in Alpe-Adria-Danube Region, RAAD, pp. 247–252.
[25] K. Lai, L. Bo, X. Ren, D. Fox, A large-scale hierarchical multi-view RGB-D object dataset, in: Proc. of the IEEE International Conference on Robotics & Automation, ICRA, 2011.
[26] K. Lai, L. Bo, X. Ren, D. Fox, Sparse distance learning for object recognition combining RGB and depth information, in: IEEE International Conference on Robotics and Automation, ICRA, 2011, pp. 4007–4013.
[27] D. Vernon, Cognitive vision: the case for embodied perception, Image and Vision Computing 26 (1) (2008) 127–140.
[28] E. Torres-Jara, L. Natale, P. Fitzpatrick, Tapping into touch, in: Proc. 5-th Intl. Workshop on Epigenetic Robotics, 2005, pp. 79–86.
[29] J. Sinapov, M. Weimer, A. Stoytchev, Interactive learning of the acoustic properties of household objects, in: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, 2009, pp. 2518–2524.
[30] A. Rebguns, D. Ford, I. Fasel, Infomax control for acoustic exploration of objects by a mobile robot, in: Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence, 2011.
[31] J. Sinapov, V. Sukhoy, R. Sahai, A. Stoytchev, Vibrotactile recognition and categorization of surfaces by a humanoid robot, IEEE Transactions on Robotics 27 (3) (2011) 488–497.
[32] H. Saal, J. Ting, S. Vijayakumar, Active sequential learning with tactile feedback, in: Proc. of the 13th Intl. Conf. on Artificial Intelligence and Statistics, AISTATS, vol. 9, 2010.
[33] L. Natale, G. Metta, G. Sandini, Learning haptic representation of objects, in: Proceedings of the International Conference on Intelligent Manipulation and Grasping, 2004.
[34] T. Bergquist, C. Schenck, U. Ohiri, J. Sinapov, S. Griffith, A. Stoytchev, Interactive object recognition using proprioceptive feedback, in: Proceedings of the 2009 IROS Workshop: Semantic Perception for Robot Manipulation, St. Louis, MO, 2009.
[35] J. Sinapov, T. Bergquist, C. Schenck, U. Ohiri, S. Griffith, A. Stoytchev, Interactive object recognition using proprioceptive and auditory feedback, The International Journal of Robotics Research 30 (10) (2011) 1250–1262.
[36] S. Takamuku, K. Hosoda, M. Asada, Shaking eases object category acquisition: experiments with a robot arm, in: Proceedings of the Seventh International Conference on Epigenetic Robotics, 2007.
[37] S. Chitta, J. Sturm, M. Piccoli, W. Burgard, Tactile sensing for mobile manipulation, IEEE Transactions on Robotics 99 (2011) 1–11.
[38] J. Sinapov, A. Stoytchev, From acoustic object recognition to object categorization by a humanoid robot, in: Proc. of the RSS 2009 Workshop on Mobile Manipulation, Seattle, WA, 2009.
[39] K. Lee, H. Hon, R. Reddy, An overview of the SPHINX speech recognition system, IEEE Transactions on Acoustics, Speech, & Signal Processing 38 (1) (1990) 35–45.
[40] D. Sun, S. Roth, M. Black, Secrets of optical flow estimation and their principles, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2010, pp. 2432–2439.
[41] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Speeded-up robust features (surf), Computer Vision and Image Understanding 110 (3) (2008) 346–359.
[42] D. Pelleg, A.W. Moore, X-means: extending k-means with efficient estimation of the number of clusters, in: 17th Int. Conf. on Machine Learning, 2000, pp. 727–734.
[43] W. Aha, D. Kibler, M. Albert, Instance-based learning algorithm, Machine Learning 6 (1991) 37–66.
[44] C. Atkeson, A. Moore, S. Schaal, Locally weighted learning, Artificial Intelligence Review 11 (1–5) (1997) 11–73.

[45] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, second ed., Morgan Kaufman, San Francisco, 2005.
[46] V. Vapnik, Statistical Learning Theory, Springer-Verlag, New York, 1998.
[47] C.J. Burges, A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery 2 (1998) 121–167.
[48] O. Chapelle, P. Haffner, V. Vapnik, Support vector machines for histogram-based image classification, IEEE Transactions on Neural Networks 10 (1999) 1055–1064.
[49] S. Zhou, Trace and determinant kernels between matrices, in: Neural Information Processing Systems, NIPS, 2004.
[50] T. Hastie, R. Tibshirani, Classification by pairwise coupling, Advances in Neural Information Processing Systems 10, 1998.
[51] L. Lam, C. Suen, Optimal combinations of pattern classifiers, Pattern Recognition Letters 16 (9) (1995) 945–954.
[52] L. Lam, Classifier combinations: implementations and theoretical issues, Multiple Classifier Systems (2000) 77–86.
[53] V. Hodge, J. Austin, A survey of outlier detection methodologies, Artificial Intelligence Review 22 (2) (2004) 85–126.
[54] J. Sinapov, A. Stoytchev, The odd one out task: toward an intelligence test for robots, in: Proc. of the IEEE International Conference on Development and Learning, ICDL, 2010, pp. 126–131.
[55] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905.
[56] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.
[57] M. Sugiyama, M. Kawanabe, Machine Learning in Non-Stationary Environments, MIT Press, Cambridge, MA, 2012.
[58] X. Zhu, Z. Ghahramani, T.J. Mit, Semi-supervised learning with graphs, Tech. Rep., Carnegie Mellon University, 2005.
[59] Z. Zhou, D. Zhan, Q. Yang, Semi-supervised learning with very few labeled training examples, in: Proceedings of the National Conference on Artificial Intelligence, AAAI, vol. 22, 2007, pp. 675–680.
[60] G. Tsoumakas, I. Katakis, Multi-label classification: an overview, International Journal of Data Warehousing and Mining (IJDWM) 3 (3) (2007) 1–13.
[61] M. Boutell, J. Luo, X. Shen, C. Brown, Learning multi-label scene classification, Pattern Recognition 37 (9) (2004) 1757–1771.
[62] R. Schapire, Y. Singer, Boostexter: a boosting-based system for text categorization, Machine Learning 39 (2) (2000) 135–168.
[63] P. Oudeyer, F. Kaplan, What is intrinsic motivation? a typology of computational approaches, Frontiers in Neurorobotics 1 (2007) 1–6.
[64] T. Gao, D. Koller, Active classification based on value of classifier, in: Advances in Neural Information Processing Systems, NIPS 2011, 2011.

Jivko Sinapov received the B.S. degree in computer science from the University of Rochester, NY, in 2005. He is currently working towards a Ph.D. degree in computer science with the Developmental Robotics Laboratory, Iowa State University, Ames, IA. His current research interests include developmental robotics, robotic perception, manipulation, and machine learning.

Connor Schenck received the Bachelor's degree in computer science from Iowa State University, Ames, IA, in 2011. He is currently working towards a Master's degree in computer science with the Developmental Robotics Laboratory, Iowa State University, Ames, IA. His current research interests include artificial intelligence, machine learning, robotics, and developmental robotics.

Kerrick Staley is currently pursuing an undergraduate degree in computer engineering at Iowa State University. He is interested in machine learning, robotics, and human–computer interaction.

Vladimir Sukhoy received the Bachelor's degree in applied mathematics from Donetsk National University, Donetsk, Ukraine, in 2004. He is currently working toward the Ph.D. degree in computer engineering with the Developmental Robotics Laboratory, Iowa State University, Ames, IA. His current research interests are in the areas of developmental robotics, human–computer interaction, computational perception, and machine learning.

Alexander Stoytchev received the M.S. and Ph.D. degrees in computer science from Georgia Institute of Technology, Atlanta, GA in 2001 and 2007, respectively. He is currently an Assistant Professor of Electrical and Computer Engineering and the Director of the Developmental Robotics Laboratory, Iowa State University, Ames, IA. His current research interests are in the areas of developmental robotics, autonomous robotics, computational perception, and machine learning.