Interactive Object Recognition Using Proprioceptive Feedback

Taylor Bergquist, Connor Schenck, Ugonna Ohiri, Jivko Sinapov, Shane Griffith and Alexander Stoytchev
Developmental Robotics Laboratory
Iowa State University
{knexer, cschenck, ucohiri, jsinapov, shaneg, alexs}@iastate.edu

Abstract—This paper proposes a method for interactive recognition of household objects by a robot using proprioceptive feedback. In our experiments, the robot observed the changes in its proprioceptive stream while performing five exploratory behaviors (lift, shake, drop, crush, and push) on 50 common household objects (e.g., bottles, cans, balls, toys, etc.). Specifically, the robot used its own joint torques recorded during each interaction to recognize the object that it was manipulating. The results show that the robot can learn to recognize objects solely from the proprioceptive information obtained while interacting with them. Furthermore, by applying multiple behaviors on the same object, the robot was able to significantly improve its object recognition accuracy. Overall, the results show that proprioception should be considered as an important source of information for object recognition and object manipulation tasks.

I. INTRODUCTION

Traditionally, most object recognition systems used in robotics have relied heavily on computer vision techniques [1, 2]. Given a clear view of an object, such systems can achieve a high degree of recognition accuracy, but still suffer from several important limitations. For example, a computer vision system cannot distinguish between a heavy object and a light object that otherwise look identical. Nor can a computer vision system recognize an object that a robot is manipulating outside its field of view. The human visual system suffers from these same limitations. Studies in cognitive psychology have repeatedly shown that other sensory modalities are necessary in order to resolve perceptual ambiguities about objects (e.g., is the object heavy or light?) [3, 4]. Hence, there is a great need to integrate other sensory modalities into robots’ object recognition models.

This paper investigates the use of proprioceptive feedback as a source of information about objects that a robot interacts with. We build upon our previous work on acoustic object recognition [5, 6] by showing that the model used for the representation of acoustic information in [5] is also very effective for representing proprioceptive feedback in the form of joint torque values. In this work, the robot interacted with 50 objects using five different behaviors (lift, shake, drop, crush, and push). The robot represented the proprioceptive information from each interaction as a sequence of state activations in a Self-Organizing Map (SOM). The SOM allows the robot to turn the high-dimensional proprioceptive information into a sequence of tokens drawn from a finite alphabet - in this case, the set of nodes in the map. The robot was tasked with recognizing the object in the interaction using proprioceptive data alone. Two different learning algorithms, designed to work on sequence data, were evaluated: Multinomial Naïve Bayes (MNB) and k-Nearest Neighbors (k-NN). The results indicate that the robot can learn to recognize objects solely from the proprioceptive information observed while interacting with them. Furthermore, by applying multiple behaviors on the same object, the robot was able to significantly improve its object recognition accuracy.

II. RELATED WORK

A. Psychology and Cognitive Science

The ways in which humans use proprioceptive information is a well-studied topic in psychology and cognitive science. Some studies have investigated human integration of proprioception with other sensory modalities, such as Sapp et al.’s study [3], in which toddlers were presented with a sponge that was deceptively painted as a rock. All of the toddlers believed that the object was a rock until the moment they touched it or picked it up. While the children could recognize the object quickly with vision alone, proprioceptive information...
was found to be necessary in order to resolve ambiguities about the object at hand.

In a similar study, Heller et al. [4] studied how a mirror could create conflicting visual and proprioceptive information. Subjects were asked to identify raised letters by touching them and viewing them through a mirror that inverted the letters vertically (e.g. a ‘p’ became a ‘b’). More than half of the time the subjects had to use proprioceptive data to correctly identify the letters. Thus, proprioceptive feedback can be more useful than vision in object recognition tasks. This implies that robots that learn to recognize objects using proprioceptive input (in addition to other modalities) would be better suited for human-inhabited environments such as our homes and offices.

Several studies have shown that both animals and humans use stereotyped exploratory behaviors to extract information about objects [7]. One study has even suggested that some birds use almost their entire behavioral repertoire to explore a novel object [8]. Presumably, a robot may also interact with objects using a stereotyped behavioral sequence in order to obtain better object recognition accuracy. It is, however, not immediately clear which behaviors a robot should utilize. This paper explores the usefulness of five exploratory behaviors for proprioceptive recognition.

B. Robotics

Object recognition is not a novel problem – it has been studied heavily in the visual domain and moderately in the auditory domain. There has been very little previous work dealing exclusively with proprioceptive object recognition. One such example is the work by Natale et al. [9] in which proprioceptive data captured from the robot’s hand was used to recognize objects. One of seven objects was placed in the robot’s hand, whereupon the hand would close until it reached a preset torque limit. The resulting joint angles on the hand were then read and fed to a self-organizing map. The robot was able to distinguish between objects of different sizes as well as between objects of similar size but different rigidity.

In other related work, Kubus et al. [10, 11] have demonstrated a method for the direct estimation of several physical properties of objects (e.g., mass and moment of inertia). These properties were then used to recognize objects as being one of three known objects [10]. It is important to note that objects rigidly grasped by the robot behave like additional links. Thus, methods for estimating dynamic models of the robot’s body (see [12, 13, 14, 15] for a representative sample) can also be applicable when estimating an object’s mass and moment of inertia. In contrast, this paper explores how a general sequential representation for high-dimensional sensory data, coupled with standard machine learning algorithms, can be used by the robot to learn to recognize the objects it manipulates. Thus, the method described here can also be applied to other sensory modalities (e.g., audition [5]).

Nakamura et al. [16] describe a robot that used proprioception along with video and audio information when interacting with objects. They investigated whether a robot could infer object properties grounded in one modality from another (e.g., whether an object would make noise when picked up after only looking at it). They found a much higher correlation between visual and proprioceptive information than between visual and auditory information.

Proprioception was used in a study by Metta et al. [17] to bootstrap a robot’s ability to work with objects. While the robot primarily relied on vision to complete its task (determining the principal axis of an object and its relation to how the object rolls), the use of proprioception also aided the robot in locating the object. This shows that proprioception can be an effective tool for use by a robot.

There has also been some work in active robot object recognition. Fitzpatrick et al. [18] studied a robot that actively learned about objects by initiating interactions with them instead of passively observing and reacting. Similarly, Natale et al. [19] have demonstrated how basic interactions can be used by a robot to learn how objects would move after a certain behavior was performed on them. Also, Takamuka et al. [20] used a robotic arm to shake nine different objects, and utilized the sounds generated by this shaking behavior to group the objects into three categories (rigid objects, paper, and water bottles).
This paper describes a method for interactive object recognition using only proprioceptive data. The method is tested using a large-scale experimental study with 50 household objects. We build upon our previous work on acoustic object recognition [5, 6] by showing that the model utilized for the representation of acoustic information is also very effective for proprioceptive information. Furthermore, we improve the object recognition model developed in [5] by allowing the robot to combine predictions from multiple behaviors on the test object in an intelligent manner.

III. EXPERIMENTAL SETUP

A. Robot

An upper-torso humanoid robot which has two 7-d.o.f. Barrett WAMs for arms and two 3-finger Barrett hands was used to perform this study. The robot is controlled in real time from a Linux PC at 500 Hz over a CAN bus interface. The raw torque data was captured and recorded at 500Hz using the robot’s low-level API.

B. Objects

The robot interacted with a set of objects, \( \mathcal{O} \), which consists of 50 common household objects, including: cups, bottles, boxes, toys, etc. (see Fig. 2). The objects were made of various substances such as metal, plastic, paper, foam, and wood. Objects were selected using three criteria: 1) they must be graspable by the robot; 2) they must not break or permanently deform when the robot interacts with them; and 3) they must not damage the robot.

C. Behaviors

The set of behaviors, \( \mathcal{B} \), consists of five exploratory behaviors that the robot performs on each object: lift, shake, drop, crush, and push. Behaviors were selected based on their expected ability to produce unique and useful information. For example, the lift and crush behaviors were expected to yield information related to the weight and compliance of the objects, respectively. The behaviors were implemented with the Barrett WAM API. Fig. 3 shows before and after images for each of the five exploratory behaviors. The raw proprioceptive data (joint torques) was recorded for the duration of each interaction (start to end). Each object was placed in (roughly) the same configuration (i.e., position and orientation) prior to behavior execution. Due to human error, however, there was still variation of the grasp contact points, as well as the contact points with the object during the push and crush behaviors across multiple trials with the same object.

IV. LEARNING METHODOLOGY

A. Feature Extraction

During the execution of each behavior, the robot records the joint torque values for all 7 joints of the left arm over the time of the interaction. The first step in the feature extraction routine is to noise filter the raw joint torque values recorded during each interaction. The dotted line in Figure 4 shows the joint torque values for J2 (shoulder joint) as the robot lifts the dumbbell object. As can be seen from the figure, the raw values are somewhat noisy and contain spike readings. To reduce this noise, the raw data is filtered using a filter of width 10 which checks for data points that lie more than 3 standard deviations away from the window median. Any such values are thrown out and replaced with the window median. The time series is then smoothed using a moving-average filter of size 10. The solid line in Figure 4 shows the resulting smoothed joint-torque values after the noise-filtering procedure is performed.

The proprioceptive feedback, \( P_i \), from the \( i^{th} \) interaction is represented as a sequence of states on a Self-Organizing Map (SOM) [21], which is one of several ways to quantize data vectors into discrete tokens or clusters. This representation is obtained as follows: let \( T_i = [t_{i1}, t_{i2}, \ldots, t_{il}] \) be the noise-filtered joint torque values for some given interaction \( i \), where each \( t_{ij} \in \mathbb{R}^7 \) denotes the torque values for all 7 joints at time step \( j \). Given a collection of joint torque records \( T = \{T_i\}_{i=1}^K \), a set of individual joint torque vectors is sampled and used...
as an input training dataset for a SOM. In other words, the
SOM is trained with input datapoints $t_{ij} \in \mathbb{R}^7$ where each data
point denotes some particular recorded joint torque values for
all 7 joints. The Growing Hierarchical SOM toolbox was used
to train a 6 by 6 SOM (i.e., 36 total states) using the default
parameters for a non-growing 2-D single layer map [22]. Due
to memory constraints, only $1/5$ of the available input data
points $t_{ij} \in \mathbb{R}^7$ were sampled at random and used for training.
Figure 5 gives a visual overview of the training procedure.
After training the SOM, each torque record $T_i = [t_{i1}^j, t_{i2}^j, \ldots, t_{i7}^j]$ is mapped to a sequence of SOM states, by
mapping each vector $t_{ij} \in \mathbb{R}^7$ to a state on the map. A mapping
function is defined, $M(t_{ij}^j) \rightarrow p_i^j$, where $t_{ij}^j \in \mathbb{R}^7$ is the input
torque vector and $p_i^j$ is the node in the SOM with the highest
activation value given the current input $t_{ij}^j$. Hence, each torque
record $T_i$ is represented as a sequence, $P_i = p_1^i p_2^i \ldots p_{l_i}^i$,
where $p_k^i \in \mathcal{A}$, $\mathcal{A}$ is the set of SOM nodes, and $l_i$ is the
temporal length of the torque record $T_i$, as shown in Fig. 6.
Thus, each $P_i$ consists of a discrete sequence over a finite alphabet. This representation reduces the dimensionality of
the proprioceptive feedback, thus affording the use of standard
machine learning algorithms designed to work on sequential
data.

B. Data Collection

Let $\mathcal{B} = \{\text{lift, shake, drop, crush, push}\}$ be the set of
exploratory behaviors available to the robot. For each of the
eight interactions, the robot performed ten trials with all 50
objects for a total of $5 \times 10 \times 50 = 2500$ recorded interactions.
During the $i^{th}$ trial, the robot recorded a data triple of the form
$(B_i, O_i, P_i)$, where $B_i \in \mathcal{B}$ is the executed behavior, $O_i \in \mathcal{O}$
is the object in the current interaction, and $P_i = p_1^i p_2^i \ldots p_{l_i}^i$,
is the sequence of most highly activated SOM nodes as the
interaction with the object unfolds.

Given this data, the task of the robot is to learn a model such
that given a proprioceptive sequence, $P_i$, the robot can estimate
the object, $O_i$, that generated the sequence. Put in another way,
given a proprioceptive sequence $P_i$, for each behavior $B_i \in \mathcal{B}$,
the robot should be able to estimate $Pr_B(O_i = o | P_i)$ for each
object $o \in \mathcal{O}$. The next section describes the two learning
algorithms used to solve this task.

C. Learning Algorithms

Two learning algorithms were used to solve the task of
object recognition: Multinomial Na"ive Bayes (a Bayesian
probabilistic model) and k-Nearest Neighbors (a lazy distance-
based learning algorithm).

1) Multinomial Naive Bayes: The first learning algorithm used in this study was the Multinomial Naive Bayes (MNB) algorithm, which falls under the family of probabilistic models. MNB is commonly used for sequence classification tasks and has found wide applicability in natural language processing, bioinformatics, and more [23].

Under the MNB model, each sequence $P_t$ is represented as a vector $d_i = (x_1^i, \ldots, x_V^i)$ of counts where $V$ is the vocabulary and each $x_i^V \in \{0, 1, 2, \ldots\}$. Each $x_i^V$ indicates the number of times word $w_t$ occurs in the sequence $P_t$. For example, if the sub-sequence $ab$ appears 50 times in the sequence, then $x(ab)^T = 50$. Given this representation, the task of the MNB model is to assign the correct object label $O_t$ given a proprioceptive sequence $P_t$. Given model parameters $Pr(w_t | O_j)$ and object prior probabilities $Pr(O_j)$, MNB computes the most likely label for a data point $d_i$ in the following way:

$$O^*(d_i) = \arg\max_j Pr(O_j) Pr(d_i | O_j)$$

$$= \arg\max_j Pr(O_j) \prod_{t=1}^{V} Pr(w_t | O_j)^{n(w_t, d_i)}$$

where $n(w_t, d_i)$ is the number of occurrences of word $w_t$ in sequence $P_t$ as specified in the feature vector $d_i$. The probabilities $Pr(w_t | O_j)$ and $Pr(O_j)$ are estimated from the available training data using maximum likelihood with a Laplacian prior (see [23] for details). To compute the feature vector $d_i$ for each sequence $P_t$, we used k-gram features with $k = 2$. Hence, the vocabulary $V$ consisted of all possible single and double letter combinations. With 36 states in the SOM, this corresponds to a feature vector of length $36 + 36^2 = 1332$.

2) k-Nearest Neighbors: K-Nearest Neighbors (k-NN) is a distance-based algorithm which does not build an explicit model of the training data [24, 25]. Instead, given a test data point, it simply finds the $k$ closest neighbors and outputs a prediction, which is a smoothed average over those neighbors. In this study $k$ was set to 3.

The k-NN algorithm requires a distance measure, which can be used to compare the test data point to the training data points. Since each data point in this study is represented as a sequence over a finite alphabet, the Needleman-Wunsch global alignment algorithm [26, 27] was used, which can estimate how similar two sequences are. While normally used for comparing biological or text sequences, the algorithm is applicable to other situations that require a distance measure between two strings. The algorithm requires a substitution cost to be defined over each pair of possible sequence tokens, e.g., the cost of substituting 'a' with 'b'. Since each token represents a state on a Self-Organizing Map, the cost for each pair of tokens was set to the Euclidean distance between their corresponding SOM states in the 2-D plane.

These two learning algorithms were selected because they utilize the sequence information in dramatically different ways. The MNB model discards nearly all temporal information, while the k-NN model utilizes this temporal information exclusively. The ultimate effect is that the MNB model emphasizes the attributes of each proprioceptive sequence, while the k-NN model emphasizes the relationship of events within that sequence.

V. RESULTS

A. Object Recognition Results

In the first experiment, the robot is tested on how well it can estimate the object in the interaction, $O_t$, given the recorded proprioceptive information $P_t$, i.e., the robot predicts the class of a novel data point, $(B_t, O_t, P_t)$, given only the torque data sequence $P_t \in \mathcal{A}^t$. Given a test data point, the robot predicts the object class, $o$, that maximizes $Pr(O_t = o | P_t)$. The performance is estimated using 10-fold cross-validation, i.e., the set of data points $(B_t, O_t, P_t)_{n=1}^N$, where $N = 2500$, is split into ten folds. During each of the ten iterations, nine of these folds are used for training the k-NN and Bayesian models and the remaining fold is used for evaluation. The performance of the model is reported in terms of the percentage of correct predictions (the accuracy) where:

$$\text{Accuracy} = \frac{\# \text{ correct predictions}}{\# \text{ total predictions}} \times 100$$

Table I shows the performance of the k-NN and Bayesian models on this task when evaluated with 10-fold cross-validation. The accuracies for each individual behavior are also shown. As a reference, a chance predictor would be expected to achieve $(1/|O|) \times 100 = 2.00\%$ accuracy (for $|O| = 50$ different objects).

Both the k-NN and Bayesian models perform substantially better than chance. Table I also shows that performance varies depending on the behavior performed on the object. Recognition is most accurate with the lift and crush behaviors, while the shake and push behaviors are much less successful. The k-NN model generally outperforms the Bayesian model, with the exception of the shake behavior, for which the two models were not significantly different. This difference in performance implies that the overall structure of the sequence is generally more important than the distribution of SOM states throughout the sequence.

The primary reason the performance varied so dramatically between the five interactions is that each interaction implicitly captured different object properties, some of which may be better suited for the task object recognition than others. For
example, the lift behavior implicitly captures the mass of the object. The crush behavior, on the other hand, indirectly captures some geometric properties as well as some information regarding the compliance of the object (e.g., the nerf football object can be compressed, while the wooden block cannot). The way in which these behaviors capture different properties of the objects implies that combining the predictions of several behaviors performed on a test object could improve the robot’s recognition accuracy.

B. Recognition using Multiple Behaviors

The robot is also evaluated on its ability to recognize objects based on data from multiple interactions. The predictions of each interaction are combined in two different fashions – one which weights the predictions of each behavior equally, and another which weights the predictions of each behavior according to the behavior’s overall accuracy. Let \( \{ P_i \}_{i=1}^M \) be a set of proprioceptive sequences generated from the same object but each coming from a different interaction (e.g., \( P_1 \) may be the proprioceptive sequence when the object is lifted, while \( P_2 \) may be the sequence when the object is subsequently dropped). In the first scenario, the robot will assign the prediction to the object class, \( o \), that maximizes \( \sum_{i=1}^M P_r(O_i = o | P_i) \). In the second scenario, given previously estimated per-behavior accuracies \( \alpha_1, \ldots, \alpha_5 \), the robot will assign the prediction to the object class that maximizes \( \sum_{i=1}^M \alpha_i P_r(O_i = o | P_i) \). The robot is evaluated by varying the value for \( M \) from 1 (the default case, in which information from only one behavior is used for prediction) to 5 (when the information from all five exploratory behaviors is utilized).

Figure 7 shows the recognition accuracy of the k-Nearest Neighbor as the number of interactions \( M \) with the test objects is varied from 1 (the default, used to generate Table I) to 5 (applying all five behaviors on the object). The blue line shows the performance of the intelligently weighted algorithm, while the red line shows the performance of the unweighted algorithm. The four dotted lines indicate the performance of each of the two algorithms given either the best or the worst possible combination of behaviors for each value of \( M \). For example, when using only 2 behaviors, performing the lift and crush behaviors achieves accuracy of around 92%, while performing shake and push achieves only around 15%. Overall, with all five behaviors, the robot’s object recognition accuracy is 93.6%.

VI. CONCLUSION AND FUTURE WORK

This paper presented a method for proprioceptive-based object recognition. The robot in this study interacted with 50 different objects by applying five different behaviors on them: lift, shake, drop, crush, and push. The robot represented the proprioceptive feedback as a sequence of the most highly activated nodes in a Self-Organizing Map. Using machine learning algorithms designed to work on sequential data, the robot was able to recognize the object in the interaction significantly better than chance. Furthermore, as multiple behaviors are performed on the objects, the robot was able to combine multiple predictions, which resulted in recognition accuracy of over 90%. The robot was also able to estimate the reliability of each behavior for the given task and, thus, weigh predictions from different behaviors accordingly, achieving an even higher recognition rate.

These results indicate that traditional vision-based object recognition systems for robots can be further improved by using proprioceptive feedback as an additional modality. In particular, incorporating other modalities is important for robots because their visual system suffers from the same
limitations as the human visual system. For example, using vision alone, one cannot tell the difference between a wooden ball and a plastic ball that look identical (e.g., painted in the same color). Hence, interactive object recognition (as opposed to passive object recognition) can be used by the robot in many situations to resolve perceptual ambiguities about objects.

There are several promising directions for future work. First, other methods for dimensionality reduction (e.g., vector quantization, or Spatio-Temporal Isomap, as used in [28]) can be applied in order to find meaningful patterns in the robot’s proprioceptive sensory stream. Another direct line for future work is to combine proprioception with information from other modalities (e.g., audio, visual movement, etc.). The representation used in this work has already shown promise when applied to audio sensory data for the tasks of object recognition [5] and object categorization [29]. Although only proprioceptive information was used in this paper, both proprioceptive and auditory sensory feedback were recorded during the data collection process. The predictions from an auditory and a proprioceptive model could be combined in order to achieve potentially greater recognition accuracy and greater robustness to environmental changes. Some preliminary results indicate that integration of audio and proprioception indeed results in an even better object recognition accuracy.

ACKNOWLEDGMENT

This work was funded in part by NSF Grant IIS-0851976.

REFERENCES