Interactive Learning of the Acoustic Properties of Household Objects

Jivko Sinapov, Mark Wiemer, and Alexander Stoytchev
Developmental Robotics Laboratory
Iowa State University
{jsinapov, banff, alexs}@iastate.edu

Abstract— **Human beings can perceive object properties such as size, weight, and material type based solely on the sounds that the objects make when an action is performed on them. In order to be successful, the household robots of the near future must also be capable of learning and reasoning about the acoustic properties of everyday objects. Such an ability would allow a robot to detect and classify various interactions with objects that occur outside of the robot's field of view. This paper presents a framework that allows a robot to infer the object and the type of behavioral interaction performed with it from the sounds generated by the object during the interaction. The framework is evaluated on a 7-d.o.f. Barrett WAM robot which performs grasping, shaking, dropping, pushing and tapping behaviors on 36 different household objects. The results show that the robot can learn models that can be used to recognize objects (and behaviors performed on objects) from the sounds generated during the interaction. In addition, the robot can use the learned models to estimate the similarity between two objects in terms of their acoustic properties.**

I. INTRODUCTION

While our sense of vision is always constrained to a particular viewing direction, our auditory sense allows us to infer events in the world that are often outside the reach or range of other sensory modalities [1]. Studies have shown that human beings have the remarkable ability to extract the physical properties of objects from the sounds that they produce [2], [3]. The importance of everyday natural sounds is perhaps best summarized by Don Norman in his book "The Design of Everyday Things":

> *"[. . .] natural sound is as essential as visual information because sound tells us about things that we can't see, and it does so while our eyes are occupied elsewhere. Natural sounds reflect the complex interaction of natural objects: the way one part moves against another; the material of which the parts are made – hollow or solid, metal or wood, soft or hard, rough or smooth." [4, p. 103]*

Most robots today, however, do not use environmental sounds as a source of information about events in their surroundings. Nevertheless, there are many situations in which such an ability would help a robot detect and reason about events in a human-inhabited environment. For example, if a robot knocks over an object that it cannot see, the sound generated by the interaction will be the primary source of information about the nature of the object. Similarly, if a human interacts with an object outside the robot's field of view, the robot will only be able to recognize the type of object and the type of interaction using the detected sound.

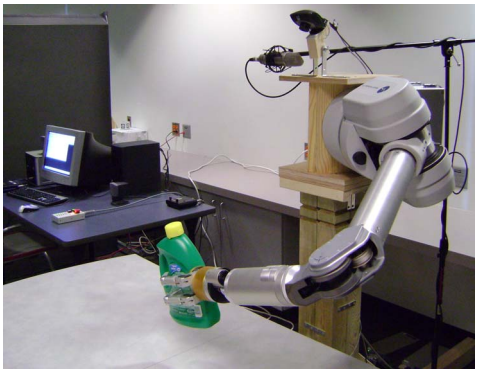

Fig. 1. The 7-DOF Barrett whole arm manipulator used in the experiments. The figure also shows the microphone used to record the sounds that different household objects make when the robot performs different exploratory behaviors on them (e.g., grasp, shake, drop, push and tap).

The problem of recognizing object interactions occurring outside of the robot's field of view presents a challenge to traditional object recognition frameworks which rely heavily on computer vision methods. Recent work in ecological psychology, however, has demonstrated that sounds generated by objects during a physical interaction with them contain structure which is often informative of various object properties (e.g., shape, size, material type) as well as the type of interaction (e.g., was the object dropped or pushed?) [5].

This paper demonstrates how a robot (shown in Fig. 1) can learn about the sounds generated by various objects through active interaction, as opposed to passive observation. The robot represents each sound as a sequence of state activation patterns through a *Self-Organizing Map* (SOM). The SOM allows the robot to turn the high-dimensional sound input into a sequence of tokens from a finite alphabet (i.e., the set of nodes in the map). Two different machine learning algorithms (k-Nearest Neighbor and Support Vector Machine) that are capable of handling data points represented as sequences were used to estimate the type of object and the interaction that generated the sound. We show how the robot can use the learned models to estimate the similarity between objects in terms of their acoustic properties.

II. RELATED WORK

There have been relatively few studies investigating how a robot could recognize objects using only auditory information. One of the earliest studies in this area was conducted by Krotkov *et al.* [6] in which the robot identifies the

material type (e.g., glass, wood, etc.) of different objects by probing them with its end effector. The results indicate that the spectrogram of the detected sound can be used as a powerful representation for discriminating between the five materials used in their study: aluminum, brass, glass, wood, and plastic [6]. Similarly, Richmond *et al.* [7] [8] have shown that modeling the spectrogram of the sounds using spectrogram averaging across multiple trials allows a robot to detect different types of materials from contact sounds.

More recently, Torres-Jara *et al.* [9] have demonstrated that a robot can recognize objects using the sounds generated when tapping on them with its end effector. After performing the tapping behavior on a novel object, the spectrogram of the detected sound is matched to one that is already in the training set which results in a prediction for the object's type. This allowed the robot to correctly recognize four different objects of varying materials.

These previous studies have used a small number of objects. In our previous work [10] we have shown that sound-based object recognition can be scaled up to a larger number of objects across multiple behaviors. Our robot used three machine learning methods (k-Nearest Neighbor, Support Vector Machine and Baysean Network) to perform object recognition on eighteen different objects by applying three different behaviors (push, grasp, and drop), using features extracted from the spectrogram of each sound. When multiple behaviors were applied to each object, the robot's recognition performance improved regardless of the learning algorithm being used [10]. The current paper uses a novel learning methodology based on temporal sequences, doubles the number of objects used in the experiments and adds two new behaviors: shaking and tapping. The novel approach outperforms the one introduced in [10] at the task of object recognition using audio information.

III. EXPERIMENTAL SETUP

A. Robot

The robot used in the experiments is a Barrett Whole Arm Manipulator (WAM) with the 3-finger Barrett Hand as its end effector (see Fig. 1). The robot has 7 degrees of freedom in the arm and 7 degrees of freedom in the hand: two per finger, and one that controls the spread of fingers 1 and 2. The robot arm is controlled in real time from a Linux PC at 500 Hz over a CAN bus interface.

B. Objects

The set of objects, $\mathcal{O}$, that the robot interacts with consists of 36 different objects, as shown in Fig. 2. The objects include common household items such as balls, cups, containers, bottles, boxes, etc. Several of the objects have contents inside of them which produce sounds when shaken. The objects are made of varying materials including metal, plastic, rubber, paper, and wood. The objects were selected from the home and office of one of the authors. The selection criteria were: the objects must be graspable by the robot, they must not contain liquids (even if they could), and they must not be fragile (i.e., no glass objects).

Fig. 2. The 36 objects used in the experiments (not shown to scale). From top to bottom: Row 1: metal box, tupperware, Vitamin C bottle, wooden cube, wooden plank, wooden stick; Row 2: plastic box, box of screws, empty shampoo bottle, box of spoons, box of thumbtacks, tennis ball; Row 3: mixed nuts jar, plastic ball, rubber ball, pop can (mt. dew), pink cup, pop can (Red Bull); Row 4: metal flange, metal plate, metal cup, box of paper clips, paper cup, pasta box; Row 5: egg carton, eraser, hard plastic bottle, plastic ball, green cup, hockey puck; Row 6: small pill bottle, coffee jar, plastic container, box of crackers, soft plastic bottle, detergent bottle.

C. Behaviors

The robot's set of behaviors, $\mathcal{B}$, consists of five exploratory behaviors that the robot performs on the objects: *grasp*, *shake*, *drop*, *push*, and *tap*. The behaviors were encoded using the Barrett WAM API. Fig. 3 shows *before* and *after* images for each of the five behaviors.

D. Sound Recording

During the execution of each behavior, the sound produced by the robot's interaction with the object was captured at 44.1 KHz using the Java Sound API over a single 16 bit channel. The microphone was a Rode NT1-A with a cardioid polar pattern having an average self noise of 5 dB and the microphone's output was routed through an ART Tube MP Studio pre-amplifier. The pre-amplifier supplied 48 volt phantom power to the microphone and sufficient gain was used on the pre-amplifier to provide a suitable input level. The recording of each sound was automatically initiated at the start of each behavior and stopped once the behavior was completed. No explicit noise filtering was performed.

The next section provides details about the audio processing, feature representation and learning methodology used by the robot to estimate the object and the type of interaction given the detected sound.

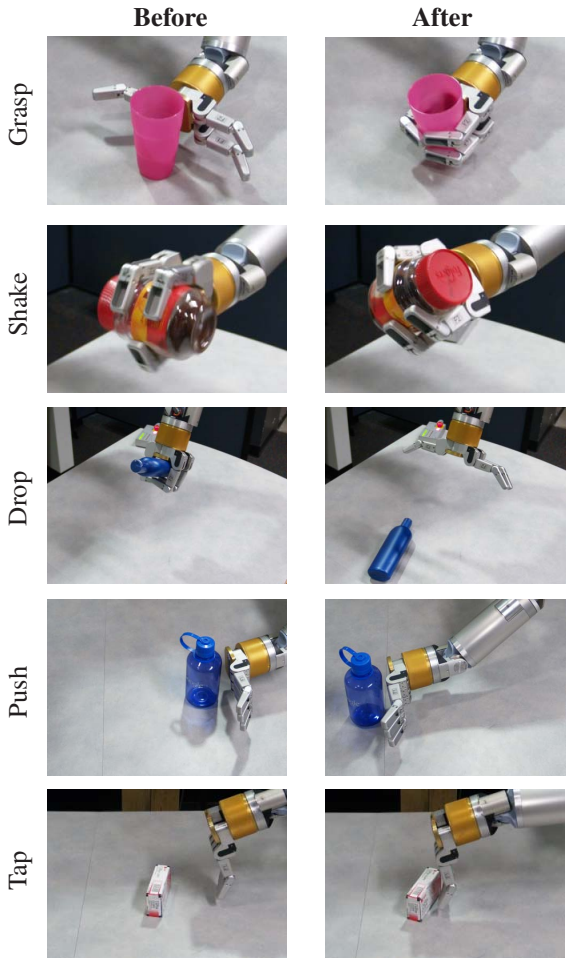

Fig. 3. *Before* and *after* snapshots of the five behaviors used by the robot.

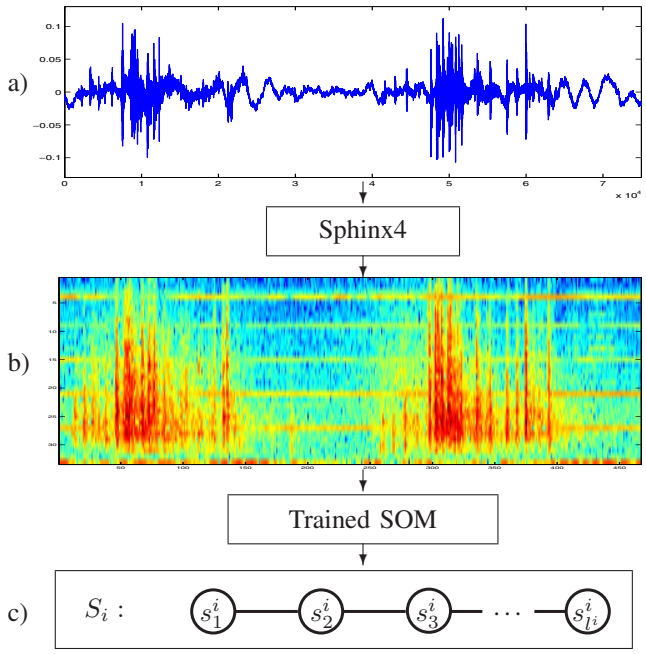

Fig. 4. Audio signal processing and sound representation: a) The raw sound recorded after the robot performs the *shake* behavior on the Vitamin C bottle. b) Computed spectrogram of the sound. The horizontal axis denotes time, while the vertical dimension denotes the 33 frequency bins. Orange-yellow color indicates high intensity. c) The sequence of states in the SOM for the detected sound, obtained after each column vector of the spectrogram is mapped to a node in the SOM. The length of the sequence S_i is l^i, which is the same as the length of the horizontal time dimension of the spectrogram shown in b). Each sequence token $s^i_j \in \mathcal{A}$, where $\mathcal{A}$ is the set of SOM nodes.

IV. LEARNING METHODOLOGY

A. Feature Extraction using a Self-Organizing Map

Each sound, S_i, is represented as a sequence of nodes in a Self-Organizing Map (SOM) [11]. To obtain such a representation, features from each sound were first extracted using the log-normalized Discrete Fourier Transform (DFT) which was computed for each sound using $2^5 + 1 = 33$ frequency bins using a window of 26.6 milliseconds (ms), computed every 10.0 ms. The SPHINX4 natural language processing library (with default parameters) was used to compute the DFT [12]. Fig. 4 a) and b) show an example sound wave and the resulting spectrogram after applying the Fourier transform. The spectrogram encodes the intensity level of each frequency bin (vertical axis) at each given point in time (horizontal axis).

Let P_i be a spectrogram, such that $P_i = [c^i_1, c^i_2, \ldots, c^i_{l^i}]$ where each $c^i_j \in \mathbb{R}^{33}$ (i.e., c^i_j is the 33-dimensional column feature vector of the spectrogram at time slice j) and l^i is the number of column vectors in the spectrogram P_i. Given a collection of spectrograms, $\mathcal{P} = \{P_i\}_{i=1}^K$, a set of column vectors is sampled from them as an input dataset and used to train a two dimensional SOM of size 6 by 6, i.e., containing a total of 36 nodes. The SOM is trained with input data points, $c^j_i \in \mathbb{R}^{33}$ which represent the intensity levels for each of the 33 spectrogram frequency bins at a given point in time. Due to memory and runtime constraints, only $1/8$ of the total available column vectors in $\mathcal{P}$, sampled at random, were used to train the SOM. The Growing Hierarchical SOM toolbox for Java was used to train the SOM [13]. The SOM was trained using the default parameters for a non-growing 2-D single layer map. Figure 5 gives a visual overview of the training procedure.

After training the SOM, each spectrogram, P_i, is mapped to a sequence of states, S_i, in the SOM by mapping the columns of P_i to nodes in the map. A mapping function is defined, $M(c^i_j) \rightarrow s^i_j$, where $c^i_j \in \mathbb{R}^{33}$ is the input column vector and s^i_j is the node in the SOM with the highest activation value given the current input c^i_j. Thus, each sound is represented as a sequence, $S_i = s^i_1 s^i_2 \ldots s^i_{li}$, where each $s^i_k \in \mathcal{A}$, $\mathcal{A}$ is the set of SOM nodes, and l^i is the number of column vectors in the spectrogram, as shown in Fig. 4. In other words, each sequence S_i consists of a sequence of states in the SOM.

The machine learning algorithms used in this study require a symmetric similarity function that can compute how similar two sequences S_i and S_j are. Computing similarity measures between a pair of sequences over a finite alphabet (e.g., strings) is a well established area of computer science, resulting in a wide variety of algorithms for exact and

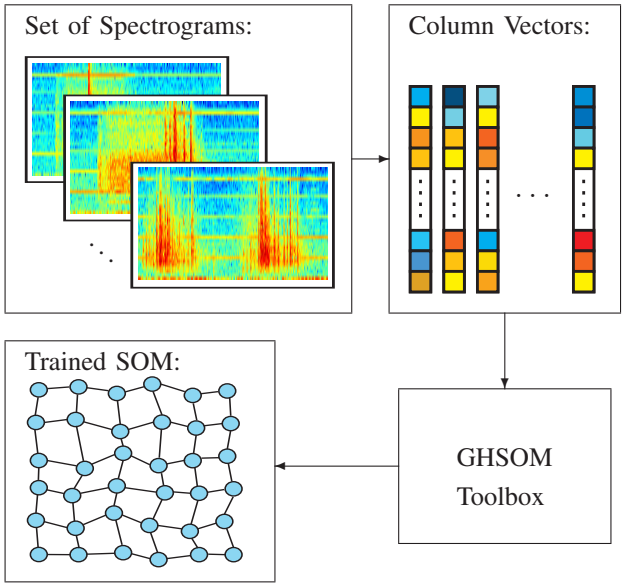

Fig. 5. Illustration of the procedure used to train the Self-Organizing Map (SOM). Given a set of spectrograms, a set of column vectors are sampled at random and used as a dataset for training the SOM.

approximate string matching [14]. In this study, we define a similarity function, $NW(S_i, S_j)$, between two such sequences to be the normalized global alignment score using the Needleman-Wunch alignment algorithm [15], [14]. While global alignment is typically used for comparing biological protein and DNA sequences, it can also be used as a measure of similarity between two sequences over a finite alphabet in other domains, such as signal and natural language processing [14]. To compute the score between two sequences, a substitution cost must be defined over each pair of tokens in the alphabet. In this study, the substitution cost between two states s_p and s_q is set to the Euclidean distance between the corresponding SOM nodes (each of which is described by its x and y coordinate in the 2-D plane) in the map.

B. Data Collection

Let $\mathcal{B} = [$*grasp*, *shake*, *drop*, *push*, *tap*$]$ be the set of exploratory behaviors available to the robot. For each of the five behaviors, the robot performs 10 trials with each of the 36 objects resulting in a total of $5 \times 10 \times 36 = 1800$ recorded interaction trials. During the i^{th} trial, the robot records a data triple of the form (B_i, O_i, S_i), where $B_i \in \mathcal{B}$ is the executed behavior, $O_i \in \mathcal{O}$ is the object in the interaction, and $S_i = s_1^i s_2^i \ldots s_{l^i}^i$ is the sequence of activated SOM nodes over the duration of the sound. In other words, each triple, (B_i, O_i, S_i), indicates that sound S_i was detected when performing behavior B_i on object O_i.

Given such data, the task of the robot is to learn a model such that given a sound sequence, S_i, the robot can estimate the object class, O_i, and the type of interaction, B_i, responsible for generating the sound S_i. In other words, given a sound S_i, the robot should be able to estimate $Pr(O_i = o|S_i)$ and $Pr(B_i = b|S_i)$ for each object $o \in \mathcal{O}$ and for each behavior $b \in \mathcal{B}$. The next subsection describes the two learning algorithms used to solve this task.

C. Learning Algorithms

1) K-Nearest Neighbor: K-Nearest Neighbor (k-NN) is a learning algorithm which does not build an explicit model of the data, but simply stores all data points and their class labels and only uses them when the model is queried to make a prediction. The k-NN model falls within the family of *lazy learning* or *memory-based learning* algorithms [16], [17].

To make a prediction on a test data point, k-NN finds its k closest neighbors in the training set. In other words, given a test data point S_i, k-NN finds the k training data points most similar to S_i. The algorithm returns a class label prediction which is a smoothed average of the labels of the selected neighbors. The normalized global alignment score, $NW(S_i, S_j)$, is used as the similarity metric between two data points S_i and S_j.

In all experiments described below, k was set to 3. An estimate for $Pr(O_i = o|S_i)$ is obtained by counting the object class labels of the k neighbors. For example, if two of the three neighbors have object class label *Plastic Cup* then $Pr(O_i = \textit{Plastic Cup}|S_i) = \frac{2}{3}$. Similarly, if the class label of the remaining neighbor is *Plastic Box*, then $Pr(O_i = \textit{Plastic Box}|S_i) = \frac{1}{3}$. The probabilities $Pr(B_i = b|S_i)$ are computed in the same manner.

2) Support Vector Machine: Support Vector Machine (SVM) classifier is a supervised learning algorithm from the family of *discriminative* models [18]. 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 $y_i \in \{-1, +1\}$. SVM learns a linear decision function $f(\mathbf{x}) =< \mathbf{x}, \mathbf{w} > +b$, $\mathbf{w} \in \mathbb{R}^n$ and $b \in \mathbb{R}$, that can accurately classify novel datapoints. The linear decision function $f(\mathbf{x})$ is learned by solving 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 [18].

A good linear decision function $f(\mathbf{x})$ in the n-dimensional input space, however, does not always exist. To overcome this problem, the labeled inputs can be 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. Rather than directly computing $\Phi(\mathbf{x}_i)$, the mapping is defined implicitly with a kernel function $K(\mathbf{x}_i, \mathbf{x}_j) =< \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) >$ which replaces the dot product $< \mathbf{x}_i, \mathbf{x}_j >$ in the dual quadratic optimization framework (see [18], [19] for details). The kernel function can also be considered as a measure of similarity between two data points.

While SVM is typically used on data points described by a numerical feature vector, it can also be applied to sequence data points. In the experiments conducted here, the kernel function between two sequences of sound data points, S_i and S_j, is defined as a power function of the normalized global alignment score, i.e., $K(S_i, S_j) = NW(S_i, S_j)^p$, where p is set to 5. Because SVM is formulated to solve a binary-classification problem, the pairwise-coupling method of Hastie *et al.* [20] was applied to generalize the original

binary classification SVM algorithm to the multi-class problem of object and behavior recognition as follows: a binary SVM is trained on data from each pair of classes (in this case objects and behaviors). During prediction, each SVM votes for one of the classes in the pair that it was trained on, and the votes are aggregated to get a final prediction. The SVM implementation in the WEKA machine learning library [21] was used, which implements the sequential minimal optimization algorithm for training the model [22].

V. EXPERIMENTS AND RESULTS

A. Recognition of Object Type

In the first experiment, the robot is tested on how well it can estimate the object in the interaction, O_i, given the detected sound S_i, i.e., the robot predicts the object class of a novel data point, (B_i, O_i, S_i), given only the sound sequence $S_i \in \mathcal{A}^{l^i}$. The performance is estimated using 10-fold cross-validation: the set of data points $\{B_i, O_i, S_i\}_{i=1}^{N}$, where $N = 1800$, is split into ten folds and during each iteration, nine of these folds are used for training the k-NN and SVM models and one fold is used for testing. Given a test data point, the robot predicts the object class, o, that maximizes $Pr(O_i = o|S_i)$. The performance of the models is reported in terms of the percentage of correct predictions (i.e., accuracy) where:

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

Table I shows the performance of the k-NN and SVM models on this task when evaluated using 10-fold cross-validation. The accuracy rates for each individual behavior are also shown. As a reference, a chance predictor would be expected to achieve $(1/|\mathcal{O}|) \times 100 \approx 2.78\%$ accuracy (for $|\mathcal{O}| = 36$ different objects).

Both, k-NN and SVM perform substantially better than chance. Table I also shows that object recognition performance can vary depending on the type of behavior performed on the object. Recognition is most difficult when the object is shaken, since most of the objects make little or no noise during that interaction. Yet for the objects that do make sound when shaken (such as the pill bottles, the mixed nuts jar, the boxes with screws and spoons, etc.), the recognition rate is near perfect for this behavior. Overall, k-NN and SVM achieve similar performance levels across all behaviors with the exception of the *grasp* behavior, for which SVM significantly outperforms k-NN, and the *drop* behavior, for which k-NN outperforms SVM.

To compare results with our previous work, the object recognition task was also performed with the sound feature representation used in [10]. When evaluated on the larger dataset used in this paper, the object recognition rates with the SVM, k-NN, and Bayesean Network classifiers (as used in [10]) were 9.33%, 11.67%, and 5% respectively, indicating that the proposed representation in the current paper is much better suited for the task. It is also important to note that the approach introduced here is more powerful than the one in [10] because it allows the robot to perform the object recognition task without explicit knowledge of the type of interaction responsible for producing the detected sound.

TABLE I
OBJECT RECOGNITION ACCURACY FOR K-NN AND SVM.

Behavior	k-Nearest Neighbor	Support Vector Machine
Grasp	67.89 %	75.27 %
Shake	49.47 %	50.56 %
Drop	85.79 %	80.56 %
Push	82.89 %	84.44 %
Tap	78.15 %	75.84 %
Average	72.84 %	73.33 %

As in our previous work [10], the robot's performance is also evaluated when given multiple sounds detected from different interactions with the object. This is particularly important in some situations because manipulation of objects (by humans and robots alike) typically involves multiple interactions performed in a sequence. Let $\{S_i\}_{i=1}^{M}$ be a set of detected sounds generated from the same object but each coming from a different interaction (e.g., S_1 may be the sound sequence when the object is grasped, while S_2 may be the sound when the object is subsequently dropped). In this scenario, the robot will assign the prediction to the object class, o, that maximizes $\sum_{i=1}^{M} Pr(O_i = o|S_i)$. The robot is evaluated by varying the value for M from 1 (the default case, in which only one detected sound is used for prediction) to 5 (when the detected sounds are from all 5 object interaction behaviors).

Figure 6 shows the recognition performance of k-NN as the robot uses multiple detected sounds (generated by different behaviors) to predict the object type. The results show that when sounds from different interactions are used, the recognition performance improves dramatically. When using the detected sounds from each interaction type (grasping, shaking, dropping, pushing and tapping), the recognition performance jumps to 99.17%.

B. Recognition of Interaction Type

In this experiment, the robot is evaluated on how well it can predict the type of behavioral interaction (e.g., grasp, shake, drop, push, or tap) responsible for the detected auditory pattern. The models are evaluated under two conditions: first, when the test data points are from interactions with familiar objects (using 10-fold cross validation), and second,

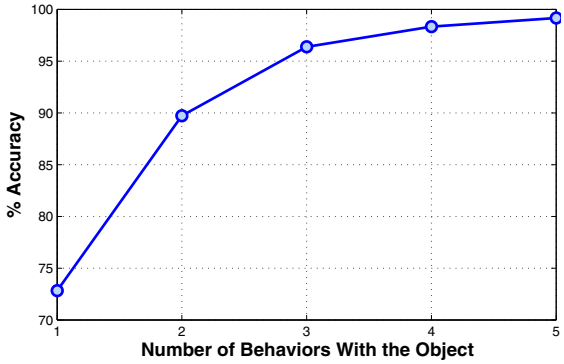

Fig. 6. Object recognition performance with k-Nearest Neighbor as the number of interactions with the object is varied from 1 (the default, used to generate Table I) to 5 (applying all five behaviors on the object).

TABLE II

INTERACTION TYPE RECOGNITION ACCURACY FOR K-NN AND SVM.

Evaluation	k-Nearest Neighbor	Support Vector Machine
Familiar Objects	99.26 %	71.74 %
Novel Objects	95.32 %	66.17 %

when the test data points are drawn from interactions with novel objects (i.e., the robot has no training data with these objects). In the second case, the cross-validation is performed such that during each iteration, the test set contains all the data points with one of the objects, while the training set contains the data points for the remaining $|\mathcal{O}| - 1$ objects.

Table II shows the results for this experiment. The first observation is that the k-NN model performs substantially better than the SVM at the task of recognizing the type of interaction responsible for the detected sound. Examination of the confusion matrix of the SVM model shows that SVM is very poor at discriminating between the *push* and *tap* interactions (i.e., SVM cannot find a good separating hyperplane between these two behavioral classes) which is why it gets a much higher error rate.

The results also show that when using k-NN, the robot can recognize the type of interactions (i.e., behaviors) performed on novel objects almost as good as in the case of familiar objects. This indicates that the sounds from different types of interactions contain distinct structure that is independent of the object used in the interaction. This could also be due to the fact that each robot behavior produces low-volume sounds generated by the robot's motors.

C. Estimating the Acoustic Similarities Between Objects

In some situations, it may be useful for the robot to have an estimate of how similar two objects are based on the sounds that they generate during physical interactions. Such an ability would allow the robot to detect objects with common physical properties (e.g., shape, size, material type) because such properties are often encoded in the sounds that the objects generate during physical interactions [5].

The robot in this study can obtain such a measure of similarity by training an object recognition model on $|\mathcal{O}|-1$ objects, and evaluating it on the one object that is left out. Let $\mathcal{C}$ be the $|\mathcal{O}| \times |\mathcal{O}|$ confusion matrix after performing cross-validation using all 1800 trials; thus, the value in the entry $\mathcal{C}_{ij}$ indicates how often object i was predicted as object j. To construct a similarity matrix between each pair of objects, let the matrix $\mathcal{C}'$ be defined such that each entry $\mathcal{C}'_{ij} = 0.5 * \mathcal{C}_{ij} + 0.5 * \mathcal{C}_{ji}$. Finally, the matrix $\mathcal{C}'$ can be turned into a distance matrix $\mathcal{C}''$ where $\mathcal{C}''_{ij} = \frac{|\mathcal{O}|-\mathcal{C}'_{ij}}{|\mathcal{O}|}$ and the diagonal of $\mathcal{C}''$ set to zeros. The entries in the distance matrix are normalized so that the results are independent of the number of objects used in the experiment. The ij-th entry in the distance matrix $\mathcal{C}''$ denotes how different two objects are based on their acoustic properties. It is difficult, however, to display such a large matrix; therefore, to visualize it, we can embed the matrix onto the 2-D plane using the Isomap method [23]. The result is shown in Fig. 7 (on the next page).

Figure 7 shows that objects with similar physical properties are often detected as being similar in terms of the sounds they generate during interactions. For example, the objects with contents inside (e.g., the box with screws, the box with thumbtacks, the jar of mixed nuts, etc.) can be seen relatively close to each other in the top middle part of the Isomap graph. In some cases, the acoustic similarity between objects is indicative of their material type - for example, the pink and green plastic cups (which are made of identical material but differ in size) are very close in the graph. Similarly, the two pop cans can be seen next to each other in the rightmost portion of the figure. In other cases, objects close to each other share common shapes, as in the case of the four balls in the lower portion of the graph. These results are consistent with the findings of Gaver [2] and Carello *et al.* [5] in psychology, who show that the structure of sounds generated during physical interactions with objects contain information about the physical properties of the objects.

VI. CONCLUSIONS AND FUTURE WORK

This paper presented a learning framework and a large-scale experimental study which investigated how a robot can use auditory information to recognize the types of objects and behavioral interactions with them. To represent each sound, the robot used a Self-Organizing Map (SOM) which allows the robot to turn the high-dimensional sound spectrogram into a low-dimensional discrete sequence over the finite alphabet of SOM states. The representation allowed the robot to use the k-Nearest Neighbor (k-NN) and Support Vector Machine (SVM) learning algorithms to solve the task of object and interaction recognition based on the detected sound. The framework was evaluated using 36 household objects and 5 different types of behavioral interactions: grasping, shaking, dropping, pushing, and tapping.

The robot was able to recognize objects with accuracy substantially better than chance, outperforming the approach proposed in our previous work [10]. The accuracy improves dramatically when the robot hears the sounds of multiple behavioral interactions with the same object. The robot was also able to recognize the type of behavior responsible for the detected sound with near-perfect accuracy using the k-NN model. The interaction recognition performance was good even when the robot was not previously exposed to the object in the interaction. These results show that sounds generated by physical interaction with objects contain information indicative of both the object and the type of interaction.

There are several lines for possible future work. First, the results in this study showed that the similarity of objects based on the sounds that they generate is indicative of their physical properties - hence, a robot should be able to use sounds to detect not only a particular object, but also some of its characteristics, such as material type, size, shape, etc. Second, the presented framework can be extended to allow the robot to detect and recognize interactions with objects performed by humans in the robot's environment. This ability would be essential for the success of the household robots of the near future.

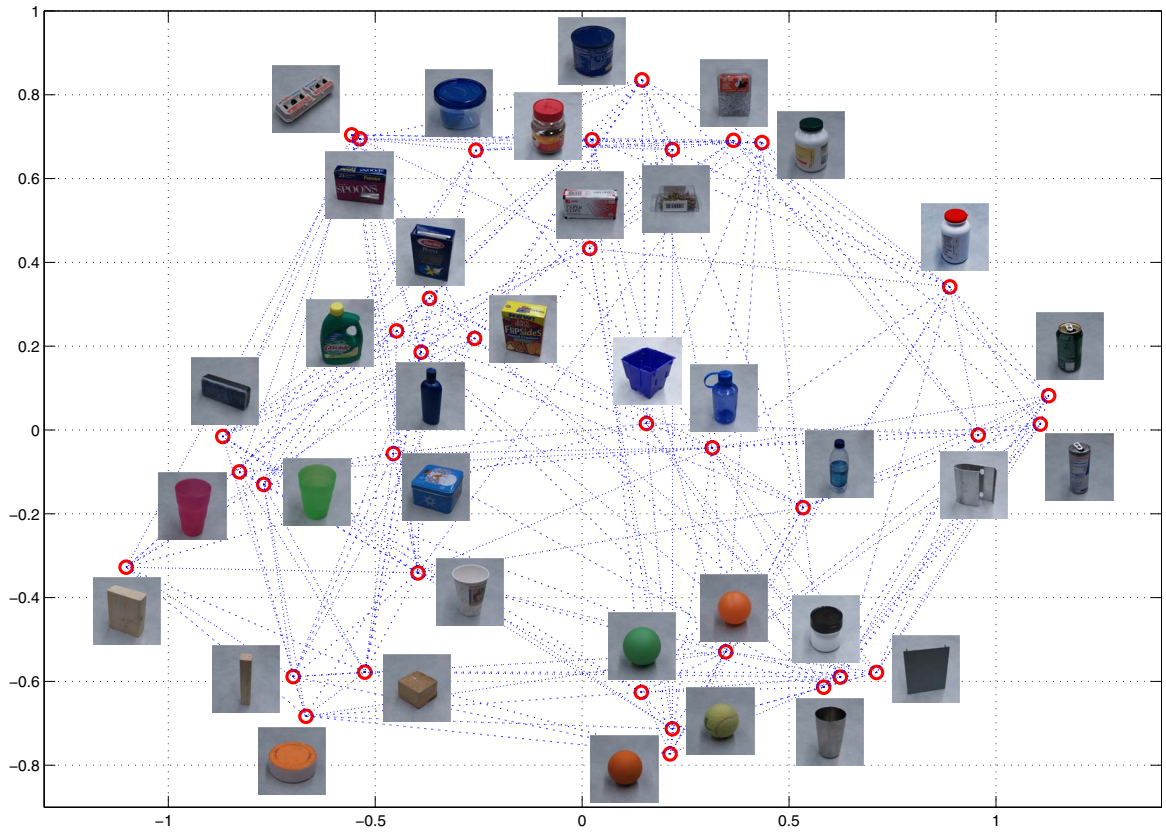

Fig. 7. Isomap visualization [23] of the learned distance matrix between the objects based on their acoustic properties. Without reading too much into the figure, we would like to point out that objects with similar physical properties (e.g., material, shape, etc.) often appear close to each other in the graph. For example, the two pop cans appear very close to each other in the rightmost portion of the graph. Similarly, the pink and green cups are very close to each other in the left portion of the graph. Also, several of the objects with contents inside of them are clustered in the top part of the graph. The Isomap method places links (shown in blue dashed lines) between each point and its k closest neighbors, where k is set to 10.

REFERENCES

[1] C. A. Fowler, "Auditory perception is not special: We see the world, we feel the world, we hear the world," *Journal of Acoustical Soceity of America*, vol. 88, pp. 2910–2915, 1991.

[2] W. W. Gaver, "What in the world do we hear? An ecological approach to auditory event perception," *Ecological Psychology*, vol. 5, pp. 1–29, 1993.

[3] M. Grassi, "Do we hear size or sound? Balls dropped on plates," *Perception and Psychophysics*, vol. 67, no. 2, pp. 274–284, 2005.

[4] D. Norman, *The Design of Everyday Things*. Doubleday, 1988.

[5] C. Carello, J. Wagman, and M. Turvey, "Acoustic specification of object properties," in *Moving Image Theory: Ecological Considerations*, J. Anderson and B. Anderson, Eds. Carbondale, IL: Southern Illionois University Press, 2005, pp. 79–104.

[6] E. Krotkov, R. Klatzky, and N. Zumel, "Robotic perception of material: Experiments with shape-invariant acoustic measures of material type," in *Experimental Robotics IV*, ser. Lecture Notes in Control and Information Sciences. Springer Berlin/Heidelberg, 1996, vol. 223, pp. 204–211.

[7] J. L. Richmond and D. K. Pai, "Active measurement of contact sounds," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2000, pp. 2146–2152.

[8] J. L. Richmond, "Automatic measurement and modelling of contact sounds," Master's thesis, University of British Columbia, 2000.

[9] E. Torres-Jara, L. Natale, and P. Fitzpatrick, "Tapping into touch," in *Proc. of the Fifth International Workshop on Epigenetic Robotics*, 2005, pp. 79–86.

[10] J. Sinapov, M. Weimer, and A. Stoytchev, "Interactive learning of the acoustic properties of objects by a robot," in *Procceedings of the RSS Workshop on Robot Manipulation: Intelligence in Human Environments, Zurich, Switzerland*, 2008.

[11] T. Kohonen, *Self-Organizing Maps*. Springer, 2001.

[12] K. E. Lee, H. W. Hon, and R. Reddy, "An overview of the SPHINX speech recognition system," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 1, pp. 35–45, 1990.

[13] A. Chan and E. Pampalk, "Growing hierarchical self organizing map (ghsom) toolbox: visualizations and enhancements," in *Procceedings of the 9th International Conference on Neural Information Processing (NIPS)*, 2002, pp. 2537–2541.

[14] G. Navarro, "A guided tour to approximate string matching," *ACM Computing Surveys*, vol. 33, no. 1, pp. 31–88, 2001.

[15] S. Needleman and C. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *J. Mol. Biol.*, vol. 48, no. 3, pp. 443–453, 1970.

[16] W. D. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithm," *Machine Learning*, vol. 6, pp. 37–66, 1991.

[17] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning," *Artificial Intelligence Review*, vol. 11, no. 1-5, pp. 11–73, 1997.

[18] V. Vapnik, *Statistical Learning Theory*. New York: Springer-Verlag, 1998.

[19] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[20] T. Hastie and R. Tibshirani, "Classification by pairwise couplling," *Advances in Neural Information Processing Systems*, vol. 10, pp. 507–513, 1997.

[21] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*, 2nd ed. San Francisco: Morgan Kaufman, 2005.

[22] S. Keerthi, S. Shevade, C. Bhattacharyya, and K. Murthy, "Improvements to Platt's SMO algorithm for SVM classifier design," *Neural Computation*, vol. 13, no. 3, pp. 637–649, 2001.

[23] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.