Learning Object Dynamics

Saket Sadani, Victoria Zhou
University of Texas at Austin
Building Wide Intelligence Lab
Autonomous Intelligence Robotics FRI Stream

Abstract

Advances in the field of cognitive science and developmental behavior has shown that infants typically learn via association and through visual-manual exploration (Johnson 2010). To mimic this developmental learning in robotic agents, the robot must be able to interact with objects and learn features as a result of its actions. Specifically, this project intends to move past using only visual cues in order to discern the characteristics and behavior of objects. In order to achieve this, we will utilize various machine learning algorithms to train the robot to be able to accurately predict future results. The process will consist of three major parts, 1) having our robotic arm manipulate different objects and recording all state information, 2) picking pertinent features to use as a basis for classification, and 3) training based on these features and then predicting the trajectory of objects depending on how the robot interacts with it. As of now, we have a small data set which we can analyze to proceed with the second and third part mentioned above.

Introduction

As humans, after our early ages, we are able to accurately discern what objects are like. This means being able to understand objects based on their shape, size, density and other such features. We can fairly easily predict what will happen to a half-full water bottle if we push it near the top and contrast that with the result of pushing it at the bottom. With our understanding of the physical world, we know that pushing at the top is likely to make it tip over (because of torque), and pushing it at the bottom will at most make it slide across the surface it's on (or perhaps do nothing at all). Results of this type obviously depend on the action taken, its relative position with regards to the object in question, and how much force is used to complete the action. Clearly, it is not possible to "hard-code" this knowledge into any physical agent as it is highly-dimensional and not feasible to encode every combination. As such, it is necessary that we explore other avenues in order to create such understanding in our, or any, robots.

Figure 1: A robot delivering a tray of food. Such a robot needs to understand the balance of the objects it is carrying, meaning that it needs to idea of the way those objects behave when certain actions are applied to them.

There are many applications for this kind of understanding. In the general case, knowing how objects behave and react to certain actions will allow a robot to make more complex decisions in a complex environment. The uses for such intelligence are widespread, and we demonstrate this variety by listing some examples of what our robot could potentially do. Consider the case where we want the robot to deliver an object to someone, as seen in the figure 1 above. It may know how to get somewhere, but it is not entirely useful if the arm cannot safely pick up an object. If we want to reach for something and there is an obstacle in the way, being able to consider the consequences of moving the obstacles without any harm would be a useful skill to have. Furthermore, understanding objects in the surroundings could allow the robot to do neat things like cleaning! The specific applications of understanding object dynamics are seemingly limitless.

Related Work

This ability to understand object dynamics is a fairly important aspect of "intelligence" for any physical agent that must interact with its surroundings. To implement this ability in

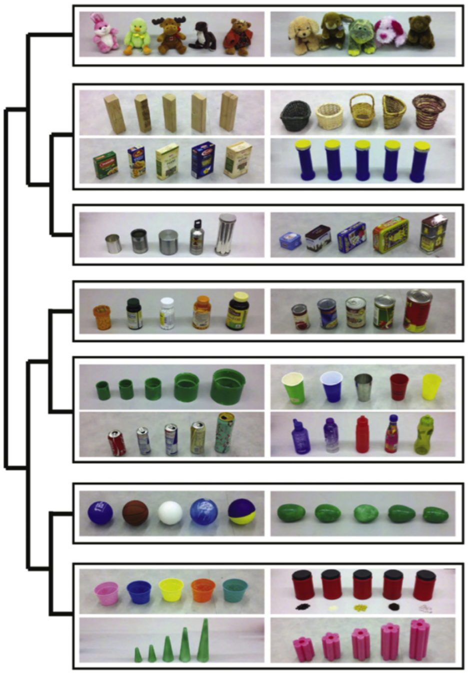


Figure 2: A hierarchical clustering of 20 categories based on the confusion matrix encoding how often each pair of categories is confused by the robot's context-specific category recognition models (Sinapov et al. 2014).

our robots in the Building Wide Intelligence (BWI) Lab, we look towards some studies in human development as well as similar robotic experiments. We see from research in cognitive science that human infants typically learn in two ways - by association and by visual-manual exploration (Johnson 2010). This second method occurs when children play with toys or objects and then gain some understanding of them. Similarly, we hope to train our infant-like robotic arm to manipulate objects, and glean data from said manipulation. If we can store data on the characteristics of the action and the resulting movements, perhaps we can predict what those actions will result in when applied in the future. In essence, can we use the results of previous iterations to guess the trajectory of subsequent ones.

Parts of our process are similar to the work seen in (Sinapov et al. 2014). This experiment attempted to classify a large group of objects into several categories based on multiple features. Sinapov et al. grouped 100 objects into 20 groups and further clustered the groups into a hierarchy as seen in figure 2. The features Sinapov et al. extracted were from visual, proprioceptive, and audio data. Proprioception takes into account the nearby parts of the body (in our case this will be the arm) as well as the relative motions of these parts. This paper makes the point that multiple sensory modalities can provide different different viewpoints and insights on certain events. As such, we too use both visual and proprioceptive data to understand the dynamics of the objects we are studying. However, as of now, we use a smaller subset of items than those used in this work.

As our data primarily results from the pushing of various objects in different ways, the idea of push affordances and object affordances came into play. Montesano and Dag show the relation between affordances and sensori-motor coordination as well as categorization of objects (Montesano et al. 2008) (Dag et al. 2010). Clearly, these are pertinent as our robot needs to understand how far something can be pushed, and how different categories of objects will behave in different scenarios.

For the complete work-flow to achieve this goal, see the Methodology section.

Methodology

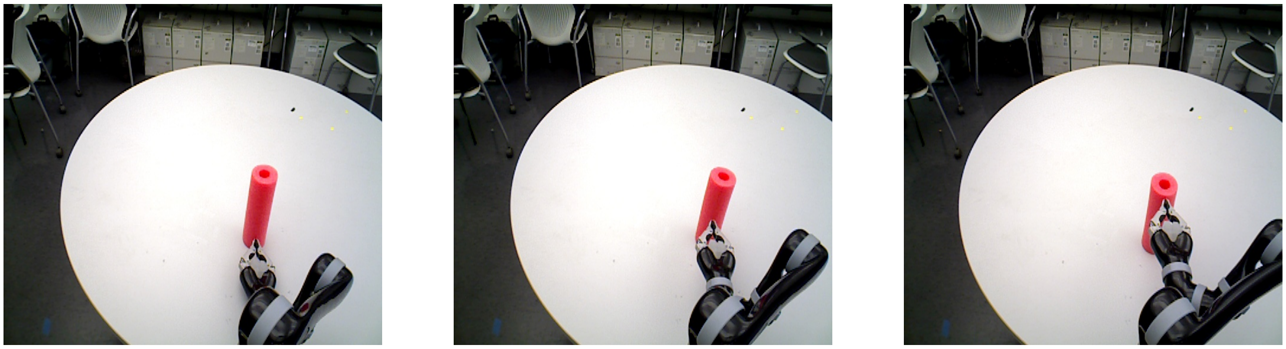
Problem Formulation

Intelligent physical agents (robots) should be able to understand objects. However, currently the agents do not yet have an understanding of the dynamics of objects, i.e. how would an object react if action A was acted upon the object. If we can teach our robots to play with objects and then learn from them that, the way infants do, then we could make our robots more "intelligent". The robot we will be using to train and collect data from is the Kinova robot arm in the Building Wide Intelligence (BWI) Lab (Larsson 1990). The arm sits on top of a Segway base. The arm itself has six degrees of freedom and has 2 gripper fingers.

To solve this problem, we designed an experiment in three stages: collecting data, extracting features, and predicting results from machine learning algorithms.

Collecting Data

We have currently collected data only for the pushing motion, although we plan to extend this soon to dropping as well as squeezing objects. In order to add robustness to our final understanding of the object, we add extra parameters to the actions. For example, we vary the angle and orientation at which the object is originally placed before testing our motion. Furthermore, we change the position on the object where the motion is performed, i.e. the top, middle, or bottom of an object if it is vertical, or different sides if its horizontal. Lastly, we applied different velocities to the object in question. The flow of this procedure can be seen below in figure 4. Examples of this are shown in figure 3. Note that though these parameters have only been enacted for the pushing action so far, they are easily extended to the other actions we want to teach the robot about.



(a) Arm height: low

(b) Arm height: middle

(c) Arm height: high

Figure 3: The push behavior was tested with 3 different heights for each object: low and close to the table as seen in 3a, near the middle of the object as seen in 3b, and at the top of the object as seen in 3c

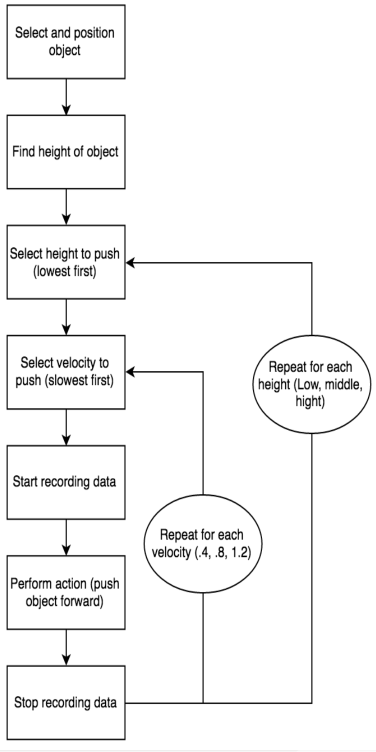

Select and position object
Find height of object
Select height to push (lowest first)
Select velocity to push (slowest first)
Start recording data
Perform action (push object forward)
Stop recording data
Repeat for each height (Low, middle, hight)
Repeat for each velocity (.4, .8, 1.2)

Figure 4: A flow chart of the experiment and actions performed on each object. For each action, each object was tested with different heights and velocities.

In Sinapov's paper, 'Grounding semantic categories in behavioral interactions: Experiments with 100 objects', Sinapov et al., trained the robot on a large data set (100 objects) with a few data points/behaviors. However, in this project we want to focus on learning the dynamics of objects and will instead be training the arm robot on a smaller set of objects (currently about 5) but with many variations of actions/behaviors (dozens of actions). In addition, the original location of the object will be kept the same to remove any noise from the background.

The raw data that we will collect will include both visual and proprioceptive data. Visual data will include the color/ RGB frames, SURF to extract the main feature points, and optical flow of before, during, and after the action is performed. The proprioceptive data collected will detail the force each joint applies during the action. In addition, the position and orientation of the arm and object before and after the action is performed, and the qualities of the action performed (the parameters on actions detailed above). These parameters and results as a whole will provide a means by which we can extract features to analyze the important components of these actions, and thus allow the robot to make informed decisions in the future.

Extracting Features

Given that the raw data exists, we want to extract the important features from the raw signals. The main questions we want to answer from the data collected are

1. Is there any change to the position or orientation of the object?
2. How far away did the object end up?
3. How long did the object take to get to it's final position?
4. What path/trajectory did the object take?

These questions can be answered by using SURF to extract the main feature points, and using optical flow of the images we collected.

The proprioceptive data collected on the forces used by each of the joints (sampled at 500 Hz) will provide many data points. Obviously, this yields too many data points to feasibly feed and train in a machine learning model. To solve this issue and extract the main features, the time (horizontal axis) will be discretized into ten temporal bins, as shown in Figure 5. A similar technique can used to extract important auditory signals (should we choose to include them) as shown in Figure 9 in the future work section.

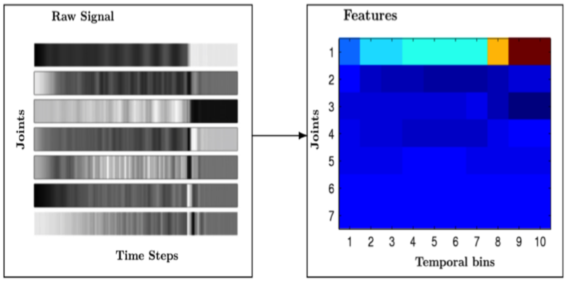


Figure 5: Illustration of the proprioceptive feature extraction routine. The input signal is sampled during the execution of a behavior at 500Hz and consists of the raw torque values for each of the robot's seven joints. Features are extracted by discretizing time (horizontal axis) into 10 temporal bins, resulting in a 7 x 10 = 70 dimensional feature vector. (Sinapov et al. 2014).

Machine Learning Algorithm and Predicting Results of Actions

Once the input and output signals have been collected, they will be fed into a machine learning models to train the physical agent to be able to predict the effects of the actions it performs on various objects, i.e. be able to predict the trajectory the object will take as the result of a push at certain angle and velocity. For example, Sinapov showed that it was possible to train the physical agent to predict and classify possible trajectories and object will take, as shown in Figure 6 (Sinapov et al. 2008).

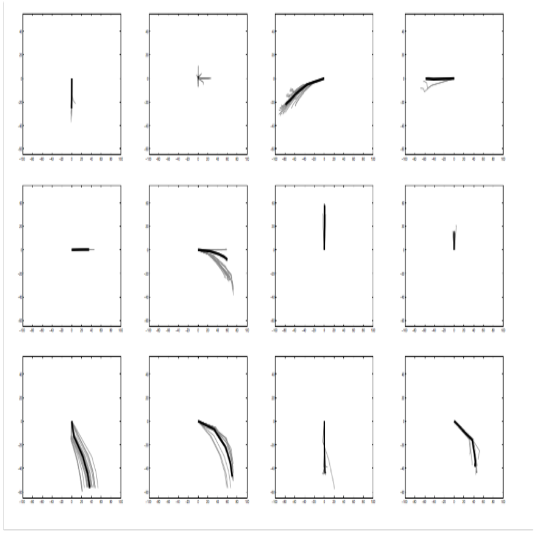


Figure 6: All twelve leaf outcome classes of the learned taxonomy for the L-Stick tool. The dark trajectory shows the outcome prototype for each leaf class in the learned taxonomy, while the lighter trajectories visualize the observed outcomes that fall within v_j (Sinapov et al. 2008).

There are several possible machine learning models that could be used to train the robot. Some examples include category recognition model, k-nearest neighbor model, decision tree learning, reinforcement learning, deep learning, and neural networks. WEKA (Waikato Environment for Knowledge Analysis) may be used as a simple and efficient way to test which machine learning models seem promising and are worth investigating further. As we further come to understand the nature of our specific inputs and outputs, we will be able to determine which machine learning will be best to use to finally train the Kinova arm to actually understand object dynamics.

Metric for Success and Evaluation

We will measure our success in three main ways: Accuracy of prediction from the machine learning model, ability to classify objects, and ability of judging behavior of new objects it has not seen before.

Machine Learning Prediction

We will want to see how accurately the robot can predict the trajectory objects will take. For example, we will compare the predicted trajectory (as mentioned above) with the actual trajectory.

Classification

We want to see how accurately the robot can classify objects into separate categories (i.e. empty box, full box, empty bottle, etc.). Some means by which this can be done are outlined in (Blum, Langley 1997), and upon utilizing these methods to classify, we can calculate our accuracy samples as follows:

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

.

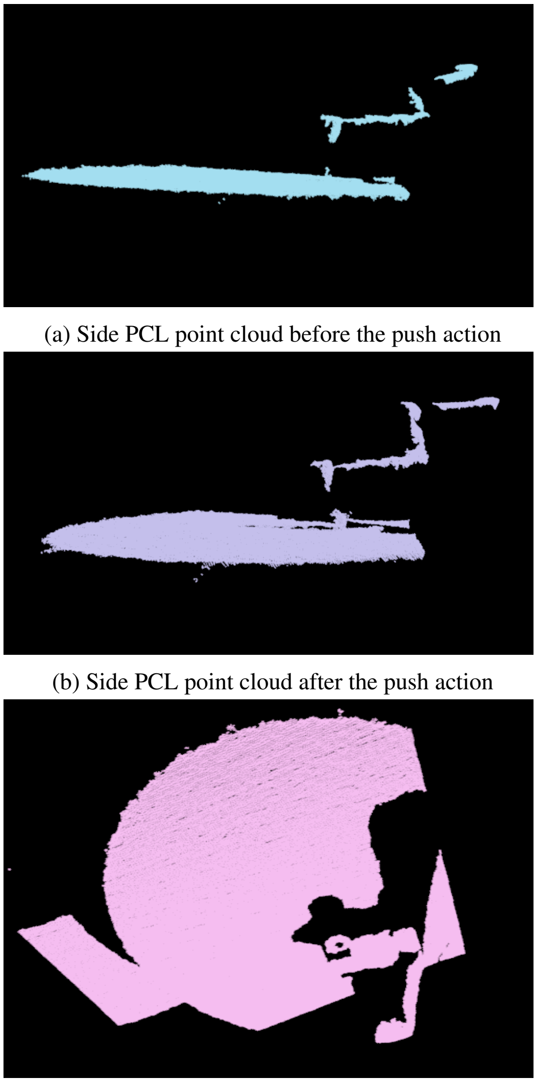
Novel Objects

Last, but not least, we will want to see the the trained robot will be able to predict the trajectories of objects and classify new objects it has not yet seen or been trained on yet. Some ways in which can utilize trajectories and our visual data can be found in (Morris, Trivedi 2008), and we are likely to combine some of those methods with (Sinapov et al. 2008)

Results and Difficulties

Data Collected

As we can see from the above Methodology section, the feature extraction and training via machine learning models are yet to take place. We have however, collected data for which to go forward with. For each action and trial performed, point clouds were taken before and after the action were performed as seen in figure 7. In addition to taking point cloud images before and after every action, color and RGB frames were taken throughout the duration of the action. An example of before and after color and RGB frames can be seen in figure 8. Besides the PCL point clouds, and RGB frames, videos from a side view were taken for all of the trials. A video of some sample trials and actions can be found here: https://www.youtube.com/watch?v=Msy-0bmPLAI.



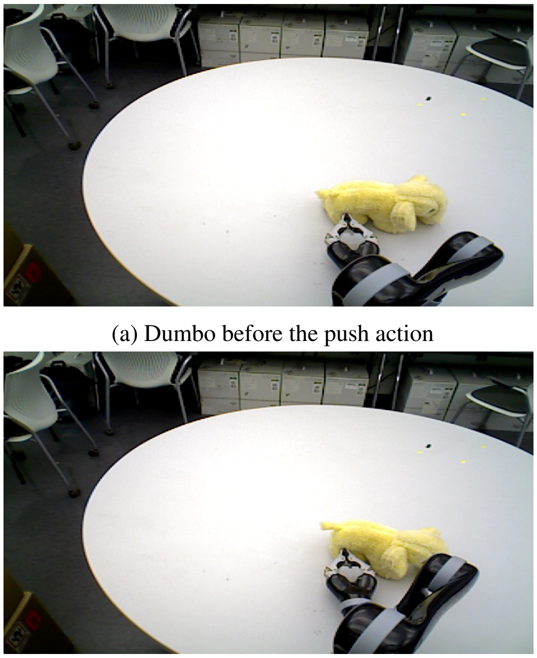
(a) Side PCL point cloud before the push action

(b) Side PCL point cloud after the push action

(c) PCL point cloud from the robot's point of view (above the object)

Figure 7: This first two set of figures depict the before 7a and after 7b of a push action with the qualities of height: high, velocity:high, push type: point. The third PCL image 7c highlights that the PCL images can be viewed from different angles: side, bottom, top (as it is in this figure).

In addition to all of the visual data collected, haptic and proprioceptive data was collected into a .csv file for each action performed. For each of the six joints on the Kinova arm, the efforts/ forced used and position were recorded. In addition, the position and orientation of the end effector, and position of the fingers of the arm were logged with the corresponding timestamp. For each action performed (over a period of 0-2 seconds), approximately 700 sets of haptic data was logged. A small sample table of the data collected can be seen in table 1 and table 2.



(a) Dumbo before the push action

(b) Dumbo after the push action

Figure 8: This set of figures depict the before 8a and after 8b of a push action on Dumbo the stuffed elephant with the qualities of pushing at the left (as a substitute for low height), low velocity (.2), and push type: point.

Difficulties

Some reasons for this delay include limited time to work on the robot arm as other groups' projects also took up significant amounts of the available time. Second, and more importantly, the Kinova arm driver seems to be a little fickle. For example, while scripting our push behavior, we ran into multiple issues in trying to issue "move to" commands in conjunction with cartesian velocity commands. The robot had trouble doing both successfully, and often even failed to accomplish each task individually. A substantial amount of time was devoted to this issue, which slowed down the data collection process significantly.

However, we do have data, and the means to collect more, so we can move forward with the next stages of our experiment.

Future Work

Continuation

There is much future work and extensions that can be done with this project. Going per the original plan, now that we have the some data collected and the means to collect more data, we should then extract the useful features from the data, feed inputs and outputs through various machine learning models, and finally testing and predicting future actions.

effort_1	effort_2	effort_3	effort_4	effort_5	effort_6
0.32103	-0.35896	0.312259	0.0320839	-0.193873	-0.115048
0.28555	-0.351233	0.333562	0.0289658	-0.193441	-0.108928
0.306941	-0.336975	0.337113	0.0371578	-0.188801	-0.113836
0.290681	-0.384471	0.351234	0.0167756	-0.19472	-0.12256
jointpos_1	jointpos_2	jointpos_3	jointpos_4	jointpos_5	jointpos_6
-0.623698	0.359012	-0.399437	-1.51844	1.20666	0.644978
-0.623698	0.359012	-0.399437	-1.51844	1.20666	0.644978
-0.623698	0.359012	-0.399437	-1.51844	1.20666	0.644978
-0.623698	0.359012	-0.399437	-1.51606	1.20666	0.644978

Table 1: A sample of the haptic data for action push on a pool noodle with the parameters: height: low, velocity: high, and push-type: hand. Efforts is the amount of force each join is using, while jointpos is the position each of the joints is in.

toolpos_x	toolpos_y	toolpos_z	toolor_x	toolor_y	toolor_z	toolor_w	finger_1	finger_2	timestamp
0.183333	0.0910539	-0.0390667	-0.49328	0.50904	0.482359	0.514668	7344	7344	1481317885
0.183333	0.0910539	-0.0390667	-0.49328	0.50904	0.482359	0.514668	7344	7344	1481317885
0.183333	0.0910539	-0.0390667	-0.49328	0.50904	0.482359	0.514668	7344	7344	1481317885
0.183629	0.0911109	-0.0390584	-0.492974	0.509316	0.482056	0.514971	7344	7344	1481317885

Table 2: A sample of the haptic data for action push on a pool noodle with the parameters: height: low, velocity: high, and push-type: hand. Toolpos is the position the end effector is in, toolor is the orientation the end effector is in, and finger_1 and finger_2 refer to whether the respective fingers are open or closed.

Extension

In addition, to continuing through with the original plan, there are many new and exciting extensions to this project. Besides just testing with a pushing action, we could also test with this with dropping, pressing, squeezing, and lifting the object. If possible, this experiment could also be generalized to more motions, not just the actions we've trained and tested on previously.

Some other extensions include using more objects for data. This way the robot has a bigger data set of knowledge to train and pull from. Furthermore, we could also add audio as a parameter for our experiment and teaching as (Sinapov et al. 2014) also used audio to help classify and categorize objects. Auditory input would be useful to distinguish, categorize, and learn the effects of an action, as the audio after an object falls has its own unique set of features. To better extract features from audio input, a discrete Fourier transform could be performed as shown in 9 by (Sinapov et al. 2014).

One of the biggest future goals of this project is to be able to get an understanding of an object based on its motion. For example, if the robot sees a bottle roll one way, it should be able to know and understand the bottle is full, versus if the bottle rolled another way. In a sense, it's doing the reverse learning and using the knowledge it has to predict and categorize the object instead of predicting the results of an action. This could be accomplished by using object affordances as (Dag et al. 2010) and (Montesano et al. 2008) did to categorize objects and to imitate actions, respectively. An affordance encodes and describes the relationship between an object, action, and result–three things that we are already keeping track of and storing.

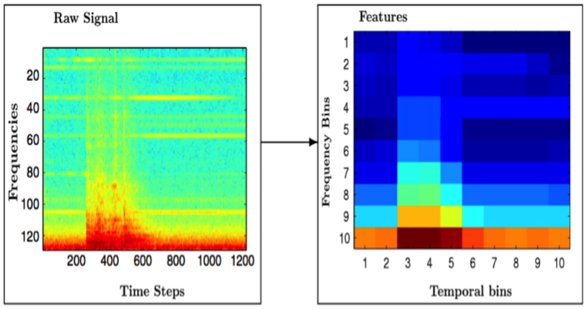


Figure 9: Illustration of the auditory feature extraction procedure. The input consists of the discrete Fourier transform spectrogram of the audio wave recorded while a behavior is executed. The spectrogram encodes the intensity of 129 frequency bins and was calculated using a raised cosine window of 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 x 10 = 100 dimensional feature vector. (Sinapov et al. 2014)

Conclusion

In essence, we attempt in this paper to each robots how different kinds of objects behave when they are acted upon in a certain way. The necessity stems from wanting capable, "intelligent" robots doing meaningful work, and the means by which we hope to enable robots to do so comes from the cognitive sciences.

Though our multi-stage project still has a ways to go, we have collected data upon which we intend to train the robot in the ways mentioned previously. Analyzing the data and the actions using our own mental heuristics, we can tell almost exactly what the robot's interaction with each object will yield (it will fall over, or roll, etc., depending on the parameters). Now, the goal is simply to allow the robot to do the same, or at least have similar predictive abilities on a small subset of actions enacted on a (for now) a small subset of objects.

Acknowledgments

Thank you Jivko Sinapov and all of the mentors in the BWI lab for giving us advice and answering all of our questions.

References

Journal Article
Johnson, S. P. (2010). How infants learn about the visual world. Cognitive Science, 34(7), 1158-1184.

Journal Article
Sinapov, J., Schenck, C., Staley, K., Sukhoy, V., & Stoytchev, A. (2014). Grounding semantic categories in behavioral interactions: Experiments with 100 objects. Robotics and Autonomous Systems, 62(5), 632-645.

Journal Article
Sinapov, J., & Stoytchev, A. (2008, August). Detecting the functional similarities between tools using a hierarchical representation of outcomes. In 2008 7th IEEE International Conference on Development and Learning (pp. 91-96). IEEE.

Journal Article
Ridge, B., Skočaj, D., & Leonardis, A. (2010, May). Self-supervised cross-modal online learning of basic object affordances for developmental robotic systems. In Robotics and Automation (ICRA), 2010 IEEE International Conference on (pp. 5047-5054). IEEE.

Patent
Larsson, O. (1990). U.S. Patent No. 4,904,148. Washington, DC: U.S. Patent and Trademark Office.

Journal Article
Ugur, E., Oztop, E., & Sahin, E. (2011). Goal emulation and planning in perceptual space using learned affordances. Robotics and Autonomous Systems, 59(7), 580-595.

Journal Article
Griffith, S., Sinapov, J., Sukhoy, V., & Stoytchev, A. (2012). A behavior-grounded approach to forming object categories: Separating containers from noncontainers. IEEE Transactions on Autonomous Mental Development, 4(1), 54-69.

Journal Article
Dag, N., Atil, I., Kalkan, S., & Sahin, E. (2010, August). Learning affordances for categorizing objects and their properties. In Pattern Recognition (ICPR), 2010 20th International Conference on (pp. 3089-3092). IEEE.

Journal Article
Montesano, L., Lopes, M., Bernardino, A., & Santos-Victor, J. (2008). Learning object affordances: From sensory–motor coordination to imitation. IEEE Transactions on Robotics, 24(1), 15-26.

Journal Article
Morris, B. T., & Trivedi, M. M. (2008). A survey of vision-based trajectory learning and analysis for surveillance. IEEE transactions on circuits and systems for video technology, 18(8), 1114-1127.

Journal Article
Blum, A. L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. Artificial intelligence, 97(1), 245-271.