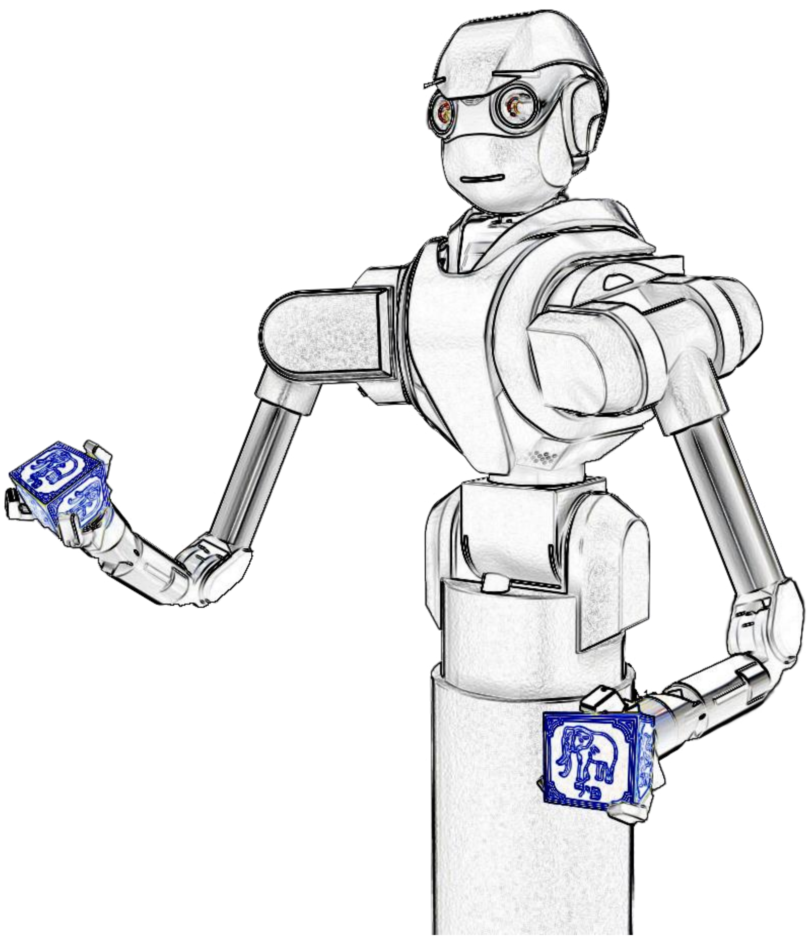
Developmentally Learning the Support Affordance of a Platform



Karl Deakyne – Yehoshua Meyer – Brian Russell

CprE 585x

Abstract

This paper describes an approach to teach a robot the support affordance of a platform. For this research, experiments will be conducted using the Developmental Robotics Lab's robot and a constructed platform. Our proposed method is composed of three stages. In the first stage, the robot will perform an exploratory pushing behavior to map the edges of the platform. Following the first stage, the robot will manipulate objects on the platform in an attempt to explore the platform edges. Using the recorded data from these two stages, machine learning algorithms will be used in order to determine the support of an object. In this case, the object can either be classified as supported or unsupported by a machine learning classifier.

Introduction

Support is a fundamental concept that humans rely on to complete a variety of rudimentary tasks. However, this concept is not understood by machines and therefore cannot be used when completing these tasks. In the modern world robots possess sophisticated hardware, but their lack of sophisticated software prevents them from autonomously performing these simple tasks. For instance, humans routinely judge how and where to place objects on a platform. In most cases, a human can solve this task with minimal effort. Humans are able to solve this task because they learn the fundamental concepts of support as infants. Learning these concepts is not a trivial task. By the age of six and half months, infants have developed a sense of whether or not an object should fall when it is placed on the edge of a platform [1]. At this age, most children are not yet capable of crawling on their bellies. Therefore, this implies that it isn't necessary to know how to walk or crawl before learning the notion of support.

Our proposed research hopes to begin working towards teaching a robot to solve the general problem by having it build its own infantile intuition of support. We will borrow accepted methods from developmental psychology which shows evidence that children learn using self-generated rules to build models of their environment [2]. Validation or violation of these rules will change a child's exploratory behaviors, and lead to new rules and an improved understanding [3]. By having the robot push a variety of blocks around (and off of) a platform, it will keep (or break) its expectations that the block will always stay on the platform. By determining the boundaries of where the expectation holds true, the robot will develop an intuition of the support that the platform affords each block. Finally, the robot will use its gained intuition to predict the support affordance boundary for a novel object.

An understanding of support is important for any intelligent being, whether it is a human or a machine. It can keep items on a table intact, determine if a potentially dangerous object will fall over, describe if an unattended object will remain where it was last seen, and even identify if an object can support an intelligent being. In the following sections of this paper we will describe more applications of our proposed work, look at related work in the fields of artificial intelligence and developmental psychology, and give a detailed explanation of how we intend to accomplish our goal.

Proposed Applications

In order to solve the grander problem of stacking objects, a better understanding of support is necessary. This research is aimed at solving the underlying task of support and is setup to teach the robot the relationship between a platform and the object(s) that it can support. Then with a more firm understanding of this relationship, future work can go into developing a way to stack one object on top of another. Because platforms can be made up of anything that is capable of support, it is easier to understand that a platform is any object that offers an affordance of support. With the ability to explore an object a robot can determine whether or not something can be supported by an object. A robot can define boundaries of an object autonomously then extend them by creating a statistical overlay of all the boundaries of a platform as it relates to the boundaries of the object that would go on top of it. As an object is placed on the platform, the boundaries of what can be placed on top of that will change. The robot could explore the new boundaries and determine where to place the next object, so that it stays within a supported area of the platform. The robot will be able to increasingly refine these boundaries since the statistical overlay gets continuously smaller with each new object.

Any time that a human has to carry or move an item, it involves using well developed notions of support. Robots will need to understand these notions as well if they are to work in a variety of environments as humans do. All objects have affordances that deal with support, and these objects can be of varying sizes and shapes. The ability that an object has to affect the support affordances of a platform can make it impractical to preprogram a robot to deal with all the possible relationships that it may be presented with. If robots are ever going to move into the workspace of people, then they are going to have to understand the support affordances of a platform just as people do.

Complicated processes that humans do will someday have assistance from robots. These processes are not the generic stacking of a box in a warehouse that never changes. These are more related to things such as, helping a moving company or carrying things throughout a lab. In situations where the platform can change or a robot will have to handle objects of different sizes and shapes, it is inefficient to store all the possible behaviors associated with every platform.

Related Work

For this research, a platform is considered an object capable of supporting another object. As infants, humans learn about platforms from watching their parents place things on tables, counters, and desks. As they age, infants are presented with toys and are allowed to explore placing these toys in a variety of places. During this process we learn to avoid placing things on the edge of a support platform. This is an important idea to understand as it helps infants avoid crawling off a cliff such as a staircase. In studies conducted by Walk and Gibson, infants were presented with a visual cliff, an apparatus that looks like a cliff's edge but is ultimately a traversable platform. In these experiments, most children were able to detect and avoid the visual cliff [4]. However, in some further studies around 66% of infants would cross to the other side of the transparent platform [5].

Some animals don't have the same issues in detecting and avoiding the "visual cliff". Gibson conducted studies with the young of many species to determine if visual cliff avoidance is innately present or if it is learned [6]. The study found that animals determine the presence of a

visual cliff by gravitating toward the shallow side of a visual cliff apparatus, showing an innate ability to detect depth. In fact, 100% of goats tested within a few hours of their birth avoided the visual cliff altogether. The differences between the goats and human trials suggested that humans are not born with an ability to detect depth; instead they have to learn the meaning of it through locomotor experience [5]. In order to gain this experience infants use models of vicarious learning and then apply it towards their own experiential play with toys.
Through this interaction they learn which objects can and cannot support something as well as what parts of the object will provide support.

Exploration is a concept that is fundamental to human learning. By examining the surrounding world, an infant begins to develop assumptions about both object and self-support. At a certain age, infants begin to discover the point at which they can support themselves [7]. However, before supporting themselves, infants begin to develop some ideas about object support [1]. Young children explore the surface of a table to determine how far a block can be pushed until it falls. This is a type of balance test that children generate to build a model of rules about surfaces. When a toy block falls off of the table it changes the child's beliefs about that surface. This leads the child to change their exploratory behaviors [2]. This type of exploratory play is how children come to terms with a surface's support affordance. While pushing a block towards the edge of a platform, a child constantly creates new rules for that platform that are based on the changes that occur during their interactions.

In the field of Artificial Intelligence, efficiency is very important, although real world problems are often perceived as trivial. In one such case known as Blocks World Planning, AI researches attempt to minimize the number of steps necessary to rearrange a stack of blocks [8][9][10]. There are many solutions to this problem, but in all of the proposed solutions one important detail is left out. How will the robots stack the blocks in the first place?

The answer to that question lies with the robot's understanding of support affordance offered by the platform. In the case of the Blocks World problem, the blocks serve as the platform. With a better understanding of where an object can be placed, placing a block on top of another would be easier for the robot to accomplish. Using this proposed extension, the Blocks World Problem can be applied to robotics as well.

In order for a robot to determine that an object is capable of being supported, it has to learn what part of a platform can offer support. To implement this we have looked into research that deals with the robot's notion of self. In a study dealing with the robot being capable of detecting itself, it has been shown that with an efferent-afferent delay system it is possible for a robot to determine that an object in its field of view is part of self, as long as it follows the same movement patterns. While moving an object, the robot can determine that the object is a part of self [11]. If the object falls off the table and is no longer moving in tandem with the robot, it can then be classified as not part of self. Using techniques that allow the robot to plot where it lost the object, our study will try to classify that object as being off the platform.

Experimental Setup

Robot Platform

The robotic platform in the Developmental Robotics Lab at Iowa State University will be used for this research. It consists of two Barrett Whole Arm Manipulators and two different Barrett Hands. One hand is equipped with tactile pressure sensors and the other is capable of vibrotactile sensory feedback with an external sensor. The robot is also equipped with two Logitech QuickCam Pro 4000 webcams for vision and two Audio-Technica U853AW cardioids microphones for auditory input. One modification of the platform will be the addition of a marked finger. The marked finger will be colored distinctively in order to track its movements in relations to the objects around it. Because of its varying sensory modalities, this robot is an excellent platform for this research.

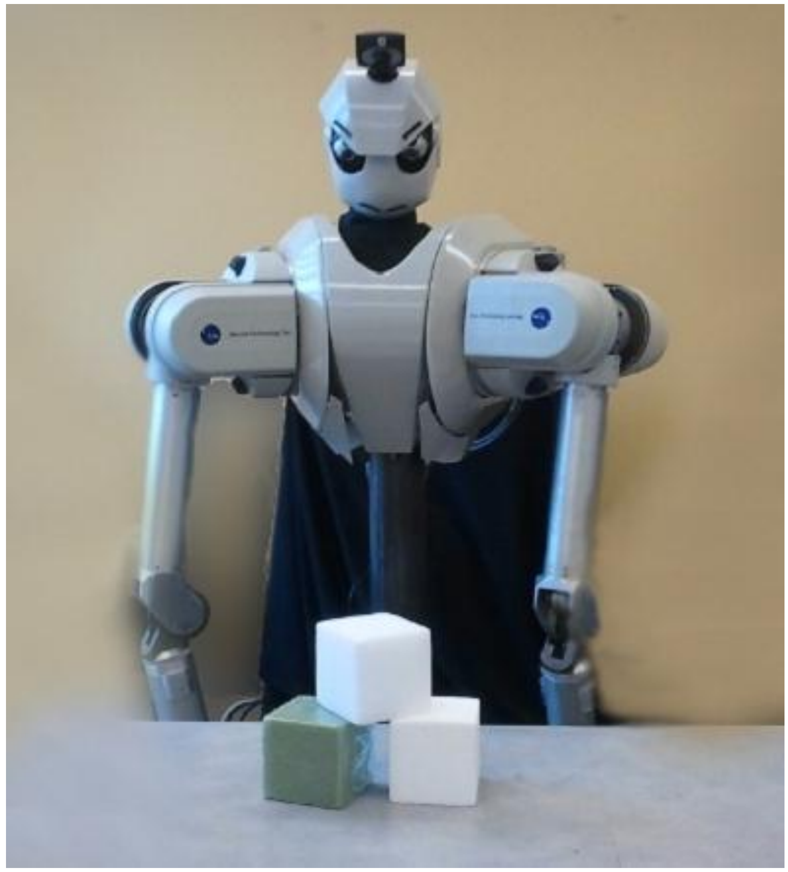


Figure 1: The Developmental Robotics Lab robot.

Data Collection

For these experiments, data will be collected during trials but will be processed offline. All modalities will be recorded in order to acquire a complete dataset. However the vision, proprioceptive, and tactile data are the only modalities that will be processed. The webcams on this platform will use the default settings with a resolution of 640x480 and a frame rate of 20 fps. For audio collection, the system will sample at a rate of 44.1 kHz in 16-bit stereo. Proprioceptive data will be sampled from the Barrett WAMs at a rate of 500 Hz and tactile data will be acquired at a rate of 15 Hz.

Proposed Constructed Platform

In addition to the robot itself, a wooden platform will be constructed in order to perform these experiments. This platform will have a single surface and will be constructed out of plywood. On the side of the platform there will be a removable ramp that will allow objects to be tracked when pushed over the edge. In this way, the robot will be able to perform the task of pushing objects off the side of the platform and self-detect itself in relation to the objects. The actual size and dimensions will be determined based on the end-effector space of the robot. A sketch of this example platform is shown in the figure below.

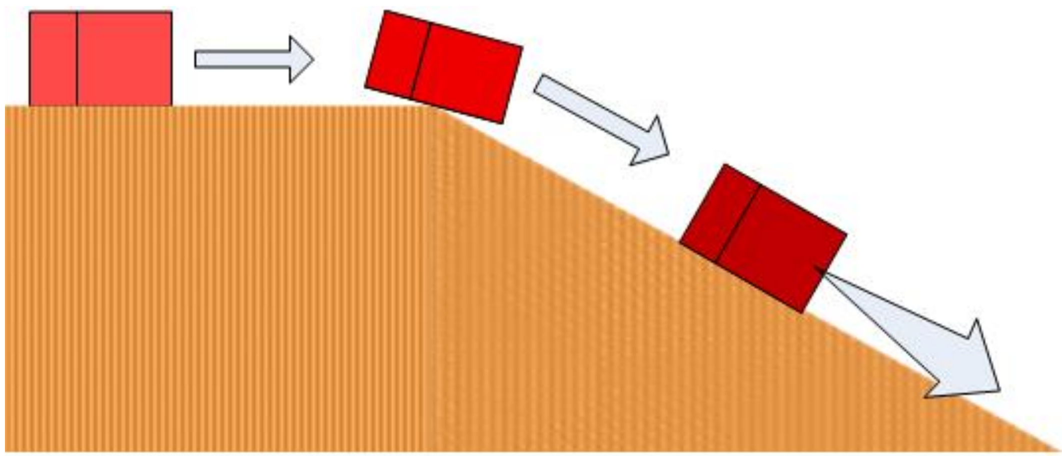


Fig 2: Illustration of an object moving across the constructed platform.

Collection of Objects

The objects for this project will be constructed from pieces of wood and will be composed of a variety of shapes. Unlike many household objects, these constructed objects will have a definite form and a uniform material density. While this may appear as oversimplification, these objects are not unlike various children's toys such as blocks and other shapes. It is anticipated that the robot should be able to learn from interactions with these objects just as children learn from manipulating blocks.

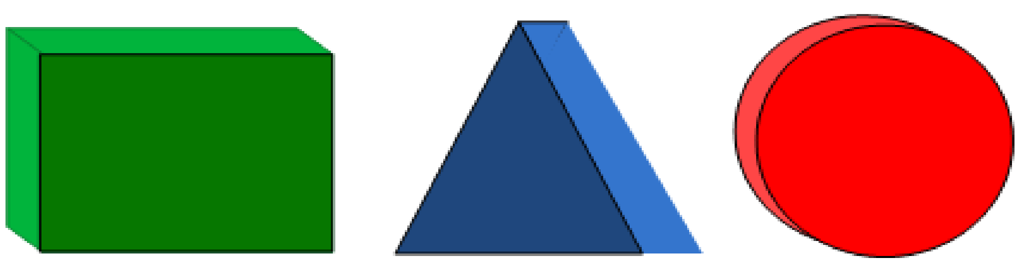


Fig 3: Examples of the shapes used in this study.

Object shapes for this experiment as still being developed and may change depending on preliminary results. However, basic shapes such as squares, triangles, and rectangles are planned for inclusion in this set. In addition to choosing shapes, distinctive colors will also be chosen in order to track the objects. While other methods of tracking will be examined, coloring the objects will ensure that color tracking is a viable method to track objects.

Software and Libraries

To bring this research to fruition, several different software packages will be used. In the dataset collection stage, the Developmental Robotics Lab's RC control software will be used to control the WAMs. Use of this software will require writing a set of scripts and plugins that will govern the robot's movement for each behavior. After creating the dataset, the OpenCV library will be used to track the objects during processing. For building models through machine learning algorithms, the Weka machine learning software implementation will be used.

Methodology

In order to learn the affordances of a platform, the robot will perform two distinct behavioral stages. The first stage will allow the robot to detect the boundaries of the platform itself. In the second stage, the robot will focus manipulating objects for the purpose of determining how they react when they are no longer supported by a platform. After the completion of these behavior stages, machine learning will be used in order to learn the affordances. The following describes this process in greater detail.

Before beginning either stage it is important to make note of the assumptions that are being made. First, it must be noted that the manipulation in the second stage will be pushing objects in one direction. Due to the time constraints of the project, it has been deemed infeasible to collect a larger dataset with a more diverse pushing behavior. Another assumption for this project deals with object tracking. In the second stage of the project, it is necessary to determine at which point an object loses the support of a platform. However, this tracking becomes difficult when objects can disappear from the visual field. Therefore, a ramp is used to effectively slow down the object during its fall. Because of the ramp, it is necessary to assume that the robot will make the same associations regardless of whether the ramp is employed or not.

The final assumption is that the robot does not have the inherent motivation that is found in people. Due to this fundamental difference, the robot must be preprogrammed to perform these exploratory behaviors. Still, there is no other known way to overcome this final assumption. With the assumptions outlined, the following paragraphs detail the implementation of this system.

Stage 1:

In the first stage of exploratory behavior, the robot will perform a babbling behavior in order to determine the platforms boundaries. During this stage, the researchers will remove the ramp portion of the platform in order to make solid contact with the robot's fingers.

When performing this behavior, the arm will start at an initial position set by the researchers and will move towards the constructed platform until it meets resistance. The boundary of the platform will be detected with this change in resistance. This will be determined by a proprioceptive event in the robot's hand. The robot will record the position for future reference in the experiment. This behavior will be performed for a finite number of sweeps, which will be determined based on the speed and range of movements that the robot is capable of. This will allow the robot to map these platform boundaries smoothly. All of the data from this stage will be used in stage two to set the boundaries of the robot's movement.

Stage 2:

At the beginning of the second stage, the robot will have a defined boundary that is based on the babbling exploration described above. This stage will focus on exploring the boundaries of the platform by pushing objects past the boundaries of the platform. In order to track the objects as they are pushed past this boundary, the ramp for the constructed platform will be put in place. Using this ramp, objects will slide at a gradual rate towards the ground instead of simply disappearing from view.

The behavior to perform this stage is as follows. First, the robot will move to an initial position with an object resting on its palm. It will then proceed to move the object forward, towards the platform's edge. Once an object is pushed past the platforms edge it will slide down the ramp at a rate different from the arm's speed.
Thus, in stage three, the hand of the robot and

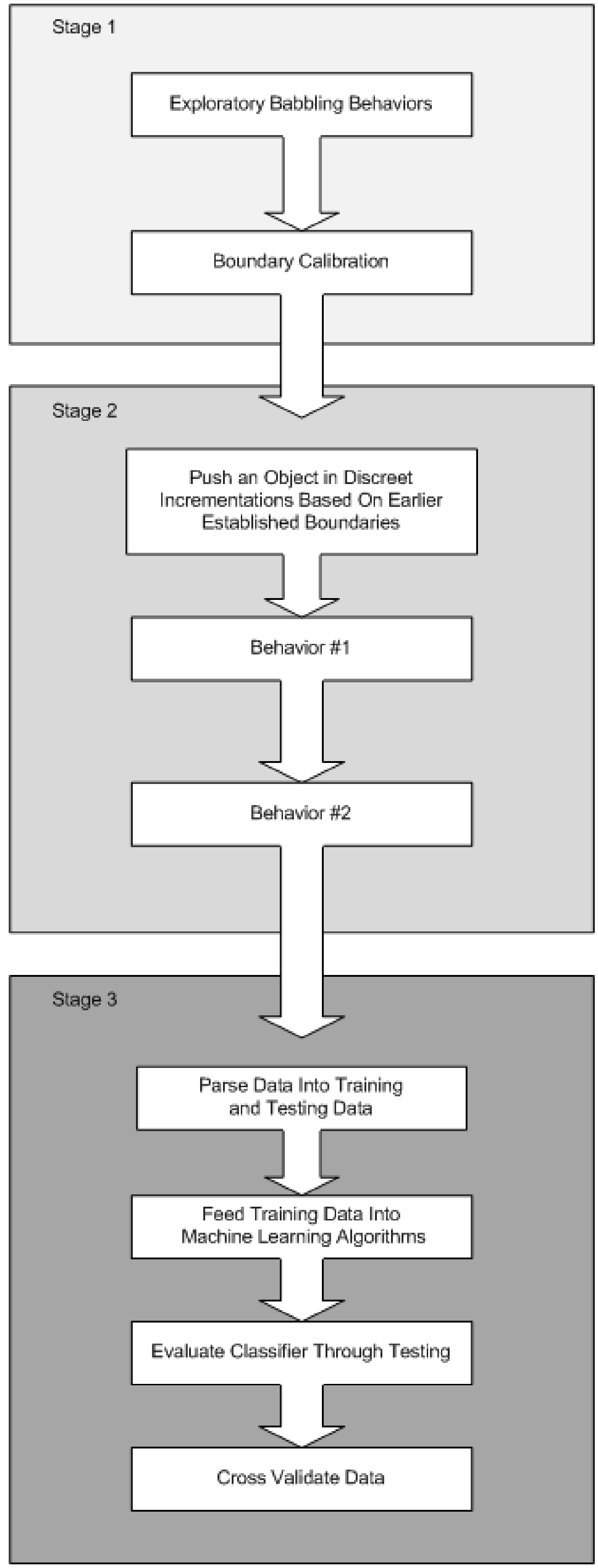


Figure 4: Diagram of implementation stages.

the object will dissociate once the edge of the platform is reached.

Stage 3:

After all object data is collected, the processing of the data will begin in the third stage. The first step in processing will be to track the objects using an OpenCV based color tracker. As previously discussed, both the hand of the robot and the object being pushed will have been marked with a unique traceable color. Therefore, the two objects will be tracked and their positions and velocities will be recorded as numerical values.

During the data gathering process, each interaction will be recorded as a tuple (o_i, d_i, m_i) where $o_i \in$ {object height, object width}, $d_i \in D$ where D is the range of distances that an object was pushed, and $m_i \in$ {movement vector of hand, movement vector of object}. For each tuple there will be a result, $S_i \in$ {support}. With this encoding scheme, this object data will be evaluated with several machine learning algorithms in order to determine the properties of an edge.

The robot will use a machine learning algorithm to determine whether or not an object is still on the platform. In order to determine if an objects in still on the table, the robot will measure whether or not the object is still moving within a set of frames. If the object is in motion, there are two scenarios. In the first scenario, the object is being pushed by the hand and therefore has a static position relative to the hand. In this case, the object also has the same velocity vector as the hand. However, in the second scenario the object is falling off of the platform onto the ramp and now has a dynamic position relative to the hand and a different velocity vector. It is this second scenario that marks the disassociation from the object being part of the arm. Furthermore, this scenario will occur after the arm reaches a point near the edge of the platform.

The machine learning algorithms used in this project are still being investigated. Initially, it was decided that a classification system would be used to associate the object with the arm's movement. Therefore, we plan to utilize algorithms such as k-nearest neighbor, Naïve Bayes, and a decision tree to classify whether the object is supported by the platform or not. These algorithms are fitting for this research as described below. In order to accelerate the development of this project, the Weka implementations of the preceding learning algorithms will be used. Weka is a Java based machine learning toolset that contains implementations of the algorithms described below [12].

The k-nearest neighbor algorithm is a lazy classifier commonly used in machine learning because of its simplicity. For this application, the k-nearest neighbor will store each of the input features and the corresponding output classifications during the training phase of the algorithm. For classification, the testing elements will sample the closest training data points and will choose a classification based on the classifications of their nearest neighbors. There are relatively few attributes from which to classify the data instances that will be collected. This algorithm is expected to perform well with the data collected and is expected to give results that can be used with the cross validation of the data.

The second machine learning algorithm used to test in this research will be the Naive Bayes classifier. For this algorithm all of the input features of the training data will be weighted probabilistically in relation to their classifications. This algorithm was chosen because it assumes all variables to be independent and identically distributed. A criterion which is met by the input data collected during stage two.

For the third machine learning algorithm, it was decided that a decision tree would be an appropriate choice. Based on the set of input training data, decision trees build a series of rules that are capable of classifying an object. Because of its availability in the Weka, the J48 (C4.5) decision tree will be used for this research. The C4.5 algorithm builds a decision tree based on the concept of information entropy [13].

Anticipated Results and Testing Procedure

For this project, we expect the robot to learn the point at which a platform loses its ability to support an object. In order to test that the robot learned this association, the researchers will use cross-validation to see if the learning algorithm can correctly find the support locations of the platform. The initial dataset will be used for cross-validation and the results will be analyzed for statistical significance. It is expected that the robot will classify the objects as supported or not supported at a rate that is significantly better than chance. If the classifier shows this then it will be understood that the robot learned this first stage of platform affordance.

Future Improvements

Successful completion of this research will warrant an extension of its principles and ideas. For future work, the robot will perform a more randomized exploration of the objects and their relation to a platform. While this work focused on pushing objects in one direction across a platform, future work will be to expand this movement. An entire platform's surface will be explored along with each of its bounding edges. Such work will improve the overall model for the support affordance of a platform.

Project Timeline

In order to complete this project within the time constraints imposed by this course, an aggressive and accelerated timetable is a necessity. The primary goals are outlined in the table on the next page. These are the tasks that will be accomplished on a weekly basis. Individual team members will have duties that are to be completed separately and together. For the tasks that can be completed separately, they have been assigned to each member after setting up the broader group tasks. The tasks to be completed as a group are more related to the robot and are setup so that there isn't one single person collecting data. It is important to have someone with the robot during the initial tests and calibrations in order to reset the position of the blocks and make sure that the robot is recording the data accurately.

Week	Goals
Week 1 (Mar 6 - 12)	Complete Project Proposal
Week 2 (Mar 13 - 19)	Build Platform, Write Control Scripts, and Collect Objects
Week 3 (Mar 20 - 26)	Gather dataset and begin analysis
Week 4 (Mar 27 - Apr 2)	Continue analysis and experiment with different machine learning algorithms.
Week 5 (Apr 3 - 9)	Complete analysis and construct a perceptual model for the affordance of a platform.
Week 6 (Apr 10 - 16)	Report results in a written report.
Week 7 (Apr 17 - 21)	Submit completed project.

Table 1: Schedule of project goals ordered by week.

During the course of each project the team's responsibilities will be divided according to previous knowledge and abilities. For the first stage of this project, Brian will build the aforementioned platform and will ensure that its specifications are compatible with the robot's feature space. Meanwhile Shuky will work on learning and implementing a color tracker in OpenCV. Karl will also be working to determine appropriate algorithms for classification.

The second stage of the project will focus on collecting a dataset from a robot and will be a joint effort for the entire group. Each member of the group will work on calibrating the robot and running the collection behaviors. For the third stage of data analysis, Brian will focus on processing the data and implementing machine learning algorithms. Karl and Shuky will aid Brian in this task by writing supporting software. In the final stage of the project, Karl will focus on reporting the results for the final report while Shuky and Brian will assist Karl in this task.

Team Qualifications

Karl Deakyne is a graduate of Iowa State University with a Bachelor's Degree in Electrical Engineering. He has basic knowledge of object oriented programming, which will enable him to write some of the code for the project while still providing a viewpoint that is drastically different from the other members of this team. Also, undergraduate courses in mathematics and probability theory have prepared him for implementing the machine learning algorithms used in this project. Other courses he has taken have developed his ability as a control systems engineer whom, combined with the Mechanical Engineering robotics course he is currently taking, should aid him in working with the robot. He is also reading several developmental psychology papers which have aided in the construction of the experimental set up for this project. As an excellent scribe, he will be the lead writer of the group and is tasked with the final edits of the group's documentation.

Shuky Meyer is an undergraduate in Computer Engineering at Iowa State. He has become proficient with C and C++ and is currently learning OpenCV and computer vision algorithms. Currently he is studying Developmental Robotics and will use what he has learned in the course to help with the material necessary for the project. He has also taken courses in statistical data analysis. Shuky has taken several courses in the fields of psychology and has had several human relations projects that will help with understanding the principles behind how humans learn the support affordance of an object through exploration of a boundary. Through reading several papers in the fields of infant cognitive development, Shuky will be tasked with making sure the setup and implementation of the experiments are within reason. Shuky's time management skills will likely be a key aspect for the organization of the group project and meetings.

As a first year Master's student at Iowa State, Brian Russell has a technical background in engineering that will be integral to the success of this project. For his undergraduate degree, he studied computer engineering and became well versed in several different programming languages. In addition to standard coursework, Brian also participated in two undergraduate research experiences where he learned the procedures to synthesize ideas into results.
Since his arrival at Iowa State, Brian has studied Artificial Intelligence and is currently enrolled in a Machine Learning course. In these classes, Brian familiarized himself with the Weka machine learning toolset. During the course of project, machine learning techniques will be used to learn the affordance of a platform. In addition to machine learning, Brian is also studying the OpenCV library in order to track the objects used in this study. Because of his previous experience with C and C++, OpenCV is rapidly being learned.

Work Cited

[1] R. Baillargeon et al., "The Development of
Young Infant's Intuition About Support." Early Development and Parenting Vol 1 (2), 69-78, 1992

[2] B. R. J. Jansen, H. L. J. van der Maas, " The Development of Children's Rule Use on the Balance Scale Task." Journal of Experimental Psychology 81, 383-416, 2002

[3] E. B. Bonawitz, S. Lim, & L. E. Schultz, "Weighing the evidence: Children's naive theories of balance effect their exploratory play." Department of Brain and Cognitive Sciences, MIT.Cambridge, MA

[4] Walk, R., & Gibson, E. A comparative and analytical study of visual depth perception, Psychological Monographs, 1961

[5] Campos, J. J., Hiatt, S., Ramsay, D., Henderson, C., & Svejda, M. The emergence of fear on the visual cliff. In M. Lewis & L. Rosenblum The origins of affect. New York: Plenum Press, 197

[6] Gibson, E. J., & Walk, R. The "visual cliff." Scientific American, 1960, 202, 64-71

[7] S. Berger, C. Theuring, K. E. Adolph, "How and when infants learn to climb stairs." Infant Behavior & Development 30, 36-49, 2007

[8] Winograd, Terry. 1971. Procedures as a
representation for data in a computer program for understanding natural language. Technical report AI TR-17, MIT Artificial Intelligence Laboratory

[9] P. H. Winston, Learning Structural Descriptions from Examples. In P. H. Winston (Ed.), The Psychology of Computer Vision. New York: McGraw-Hill, 1975. pp. 157-209

[10] John Slaney, Sylvie Thiebaux, Blocks World revisited, Artificial Intelligence, Volume 125, Issues 1-2, January 2001, Pages 119-153

[11] Stoytchev, A., "Behavior-Grounded Representation of Tool Affordances," In Proceedings of IEEE International Conference on Robotics and Automation (ICRA), pp. 3071-3076, Barcelona, Spain, April 18-22, 2005

[12] I. H. Witten and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005

[13] Quinlan, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993