Curiosity-driven learning of traversability affordance on a mobile robot

Emre Uğur, Mehmet R. Doğar, Maya Çakmak, Erol Şahin
Kovan Research Lab., Dept. of Computer Eng.
Middle East Technical University, Ankara, Turkey
{emre,dogar,maya,erol}@ceng.metu.edu.tr

***Abstract*— The concept of affordances, as proposed by J.J. Gibson, refers to the relationship between the organism and its environment and has become popular in autonomous robot control. The learning of affordances in autonomous robots, however, typically requires a large set of training data obtained from the interactions of the robot with its environment. Therefore, the learning process is not only time-consuming, and costly but is also risky since some of the interactions may inflict damage on the robot. In this paper, we study the learning of traversability affordance on a mobile robot and investigate how the number of interactions required can be minimized with minimial degradation on the learning process. Specifically, we propose a two step learning process which consists of bootstrapping and curiosity-based learning phases. In the bootstrapping phase, a small set of initial interaction data are used to find the relevant perceptual features for the affordance, and a Support Vector Machine (SVM) classifier is trained. In the curiosity-driven learning phase, a curiosity band around the decision hyperplane of the SVM is used to decide whether a given interaction opportunity is worth exploring or not. Specifically, if the output of the SVM for a given percept lies within curiosity band, indicating that the classifier is not so certain about the hypothesized effect of the interaction, the robot goes ahead with the interaction, and skips if not. Our studies within a physics-based robot simulator show that the robot can achieve better learning with the proposed curiosity-driven learning method for a fixed number of interactions. The results also show that, for optimum performance, there exists a minimum number of initial interactions to be used for bootstrapping. Finally, the trained classifier with the proposed learning method was also successfully tested on the real robot.**

I. Introduction

The concept of affordances was introduced by J.J. Gibson to explain how inherent "values" and "meanings" of things in the environment can be directly perceived and that how this information can be linked to the action possibilities offered to the organism by the environment [1]. In this sense, a stone affords throwing, a flat rigid surface affords walking, etc. Moreover, their affordances are directly perceived by humans without creating object models with further "mental calculation" of the otherwise meaningless perceptual data. This point of view also entails economical usage of the perceptual resources.

Although J.J. Gibson made a number of references to the learning of affordances, his main interest was into the perceptual aspect. However, the issue of learning of affordances have attracted attention from both psychologists and roboticists, and these studies will be briefly presented in the next section. In the rest of the paper, we will present our approach to learning of affordances.

A. Learning Affordances

E. Gibson studied the learning of affordances in humans in a developmental framework. She argued that learning is neither construction of representations from smaller pieces, nor association of a response to a stimulus. Instead, she claimed that learning is "discovering *distinctive* features and *invariant* properties of things and events" [2], "discovering the information that specifies an affordance" [3]. She pointed out that babies use exploratory activities, such as mouthing, listening, reaching, shaking, to bring about "information about changes in the world that the action produces" [2]. She suggested that, as development proceeds, exploratory activities become performatory and controlled, executed with a goal.

The problem of learning affordances has recently been studied also within autonomous robotics. These studies which dealt with the learning of various types of affordances mainly tackled two major aspects of the problem. In one aspect[4], [5], [6] the invariant properties of the environment that afford a certain behavior is learned. In the other studies,[7], [8], [9] the affordance learning is referred to as the learning of consequence of a certain action in a given situation. In [7], the robot learns what it can do with an object (e.g. rolling) only by acting (e.g. tapping or pushing away) on it, and observing the effects in the environment. In [8], [9], Stoytchev et. al. studied the so-called 'binding affordances' and 'tool affordances', where learning binding affordances corresponds to discovering the behavior sequences that result in the robot arm binding to different kinds of objects, whereas learning tool affordances corresponds to discovering tool-behavior pairs that give the desired effects. Although these studies are important in learning through exploration, in both studies, the objects are differentiated using their colors only, and no association between the visual features (that affect the affordances) of the objects and the corresponding affordances are established, giving no room for the generalization of the affordance knowledge for novel objects. Fritz et al. [10] also demonstrated a system that learns to predict the lift-ability affordance for different objects, where predictions are made based upon features of object regions extracted from camera images.

B. Curiosity-driven learning

In robotics, learning is a costly process. Ideally, the robot should physically interact with its environment exploring its environment and testing its behavioral abilities in different situations. Even for simple tasks, such as avoiding objects, a large number of interactions, some of which may result in physical damage to the robot, need to be carried out to drive the learning process. Hence, the learning process is not only time-consuming and costly in terms of the physical wearing out of the robot, but is also risky, since some of the interactions may result in physical damage to the robot. Therefore, it is essential that the interactions of the robot during the learning phase be minimized with minimal or no degradation of learning.

The problem of selection of the best training data to increase the performance and speed of learning has been studied in the field of Machine Learning (Active Learning) and particularly in Developmental Robotics. In these studies, as stated in [11], generally two modules are used: the *learner* and the *meta-learner*. In these systems, the *learner* is responsible from the learning process, whereas *the meta-learner* is responsible from selection of the next sample, which would increase the speed of the learning process. In this paper we will not use a *meta-learner* but we will utilize a curiosity-based scheme on the *learner* itself to increase the speed of the affordance learning and minimize the number of interactions with minimal degradation in learning process.

II. The KURT3D robot platform and the traversability problem

Kurt3D is a medium-sized ($45cm \times 33cm \times 47cm$), differential drive mobile robot, equipped with a 3D laser range finder. The 3D laser scanner is based on a SICK LMS 200 2D laser scanner, rotated vertically with an RC-servo motor. The 3D laser scanner has a horizontal range of $180°$, and is able to sweep a vertical range of $\pm 82.8°$ generating a 720×720 range image in approximately 45 seconds.

Kurt3D is simulated in MACSim[12], a physics-based simulator, built using ODE (Open Dynamics Engine),an open-source physics engine. The sensor and actuator models are calibrated against their real counterparts. The leftmost part of Fig. 1 shows a scene from the simulator.

A. Traversability

In this paper, we study the learning of traversability affordance for the KURT3D robot platform. The verb "traverse" is defined as "to pass or move over, along, or through". The environment is said to be traversable in a certain direction, if the robot (moving in that direction) is not enforced to stop as a result of contact with an obstacle. Thus, if the robot can push an object by rolling it away, that environment is said to be traversable even if the object is on robot's path, and a collision occurs. Hence, the traversability affordance for a robot highly depends on the location, orientation, and shape of the objects in the environment. The robot should be able to perceive the features of the environment related to the traversability affordance, in order to learn these affordances and use them in control.

The traversability problem has also recently been studied in [13] for outdoor navigation problem, where low level features, which are extracted from stereo-vision and texture based methods, are used in learning and predicting of affordances of outdoor objects. The proposed architecture supported on-line learning of the traversability affordances in unknown environments, and enabled successful execution of path plans while adapting to a completely unknown environment.

In a previous work of ours [14], we studied the learning of traversability affordance of KURT3D in simulated environments that are cluttered with objects with different shapes and arbitrary sizes, thus with different affordances:

- rectangular boxes ( ) that are non-traversable,
- spherical objects ( ) that are traversable since they roll in all directions,
- cylindrical objects, either in upright position ( ) (non-traversable), or lying on the ground ( ) (traversability depends on their orientation with respect to the robot).

In that work, we had gathered the training data from 3000 interactions of the simulated KURT3D robot in MACSim. This data was then used to learn the traversability affordance for the robot in a batch mode, and the results were then successfully transferred to the real robot.

III. Robot Control System

The robot is provided with seven simple hand-coded actions, which result in movement in seven different directions. One of the actions makes the robot go forward, while the others first rotate the robot around its own vertical axis for a certain period and then drive it forward. Along with each action, the expected displacement of the robot is provided as its success criteria.

A. Perception

The robot makes a 3D scan of the environment to obtain a range image. As shown in right part of Fig. 1, first, the image is down-scaled to reduce the noise. Then, it is split into uniform size rectangular grids and a number of distance and shape related features are extracted for each grid.

The distance related features are chosen as the distances of the closest, furthest, and mean distances of the grid. The shape related features are computed from the normal vectors of the surfaces that are computed from the range image. The direction of each normal vector is represented using two angles φ and θ, in latitude and longitude respectively and two angular histograms are computed. The frequency values of these histograms are used as the shape related features.

B. Learning

In the learning phase the robot learns a mapping between environmental situations and the results of its actions, by physically interacting with the environment. In each interaction episode, the robot is placed at a random position and orientation in a training room which includes a number of randomly placed objects.

After the robot perceives its environment using the 3D range scanner and computes a feature vector, the *learner* that is trained upto that point determines whether the current situation is an interesting one or not, based on the computed feature vector. If the learner is certain about the effect to

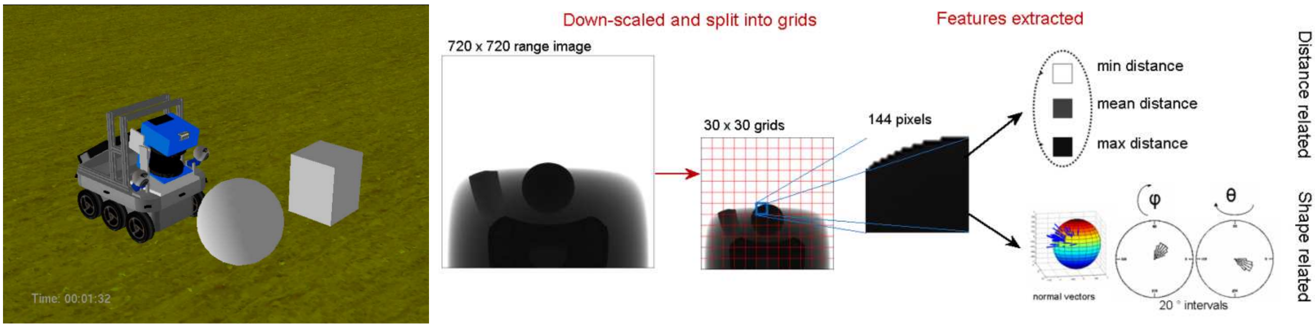


Fig. 1. The simulated robot in MACSim is shown on the left. The range image obtained in this situation and the operations applied to this image are shown on the right. The 360×360 pixel range image is divided into $30 \times 30 = 900$ grids of 12×12 pixels, and the angular histogram is divided into 18 intervals, so that total number of features computed over a downscaled range image is $900 \times (3 + 2 \times 18) = 35100$ where 3 corresponds to the three distance values (minimum, maximum, and mean) and the multiplication by 2 corresponds to the two angle channels.

be produced, the robot will choose not to interact with the environment to test its hypothesis and will be "beamed" to a different position in the room. However, if *the learner* is not certain about the result of executing a particular action in that situation, the robot will execute the action and observe the result of that action using a pre-defined success metric (displacement vector). Then, *the learner* is updated using the feature vector and the result of the action.

The learning process consists of two phases:

1) Bootstrap phase: In this phase, a small set of training samples ($n_{bootstrap}$) are obtained by interacting with the environment without any novelty check. Since time and space requirements of learning from samples with 35100 features would be huge, the learning is done using only a subset of these features. This subset includes the features which are relevant for a particular action, and affordance learning for that action is performed using only that subset.

ReliefF algorithm[15], which estimates the relevance of each feature based on its impact on the target category (traversable/non-traversable) of the samples, is used for feature selection. After computing the relevances using ReliefF, the most relevant n features are chosen. Although ReliefF does not work optimally with such a small sample set and high number of features, by setting n to a relatively large number, most of the relevant features would be included in the obtained subset. We set n to 2500 in our experiments.

The bootstrap period is also required to initiate the training. Thus, the set of training samples, obtained in this phase are used to train a classifier in a batch manner. The details of the classifier, which learns a relation that maps the (initially perceived) relevant features to predict the success/fail result of applying that action, will be given below.

2) Curiosity-driven learning phase: Different from the approaches mentioned in Section I, we will use *the learner* both to select the next sample and to learn from experience. A training sample in our domain is obtained through perceiving the environment, physically interacting with it, and storing the perceptual data together with the result of the robot's interaction (afforded/not-afforded). Thus, if *the learner* decides that a candidate sample is not interesting enough, it will not be included in training. In this case, there

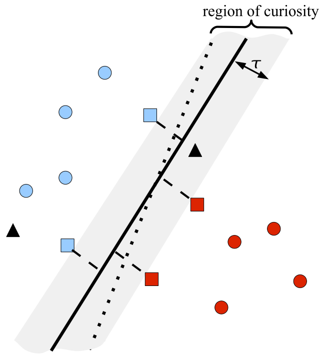


Fig. 2. The mechanism which selects interesting samples for training is demonstrated. The continuous line demonstrates the separating plane that is constructed so far, the square shaped samples demonstrates support vectors, and the circular shaped ones show the samples used in previous training steps, but not serve as support vectors. The triangular shaped samples are the candidates whose classes (traversable/non-traversable) are not known. Current SVM is more certain about the class of the sample on the left, so this candidate will not be included in the training set. However, the candidate on the right is very close to the hyperplane and SVM is not certain about its class, thus it will be included in training. A probable modification in the hyperplane is shown with dashed line after SVM is updated with this candidate sample.

is also no need to execute the action since only the perceptual data are used by *the learner* to determine whether that sample is interesting or not. In [14], a batch learner was employed which stores all training samples beforehand. In our case, a batch learner would enforce execution of all actions, even uninteresting ones during exploration phase. As a result, we should use an online-learner, which first determines the novelty of the perceptual data, and executes the action only if the perceptual data are interesting enough for that action.

Support Vector Machines (SVMs) are used to learn the mapping between perceptual data and affordance classes (traversable/non-traversable). In SVMs, the optimal hyperplane that separates two classes is found, based on the most informative samples (the support vectors) in the training set. The new test sample's class is predicted based on its relative location with respect to this hyperplane in the feature space. We made an assumption that SVMs are more certain in their

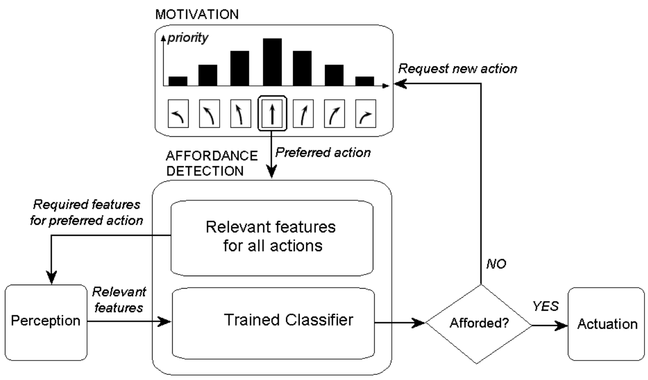


Fig. 3. Usage of the trained affordance classifiers.

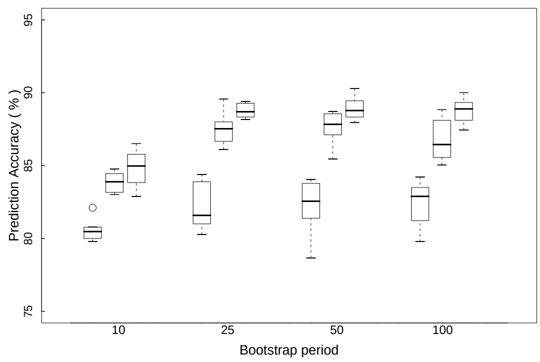


Fig. 5. The bootstrap period, which is required to select the relevant features and train an initial learner is adjusted, and the speed/performance plot is demonstrated. Curiosity paramter τ is fixed to 0.5.

class prediction of a new sample, if that sample is further away from the hyperplane, and less certain if sample is closer to the hyperplane. Thus, when the robot is in an environment, where it is almost certain about the affordances provided, it will bypass this environment without executing any action, and look for more novel situations. On the contrary, when the robot encounters a new situation, if the feature vector computed in that situation is close to the hyperplane, SVM will conclude that this situation is interesting enough to be included in training. In this case, the robot executes the action, and SVM is updated using the feature vector and the result of that action. Thus, the novelty of the candidate is determined based on its distance to the hyperplane that is constructed so far. If the distance is smaller than a fixed threshold τ then the sample is considered as an interesting one, if it is bigger than τ, it is skipped. Fig. 2 provides a simple and clear demonstration of the idea.

Although, SVMs are used as batch learning systems in general, some online implementations, where the samples are fed to the learning system in an incremental manner, are able to produce similar results. We used the LASVM software [16] for online updating of the SVM and making predictions on the candidate and test samples. A linear kernel (with tolerance parameter set as 1) was used since more complex kernels did not increase the performance in our case.

C. Control

The robot is driven using a simple control system (Fig. 3), which utilizes learned relevant feature perception and affordance classification schemes explained in the previous sections. Whenever a new action is requested, the motivation based control system sets a new *preferred* action with highest priority, among a set of actions with fixed priorities. The features which are relevant to the *preferred action* are then requested from perception, and these features are supplied to the trained classifier (SVM) to predict whether this action is afforded or not. If the immediate environment does not afford this action, a lower priority action is requested from the motivation module. Otherwise, it is executed (robot moves in a certain direction for a certain duration), and a new action is requested upon the completion of the action.

IV. EXPERIMENTAL RESULTS

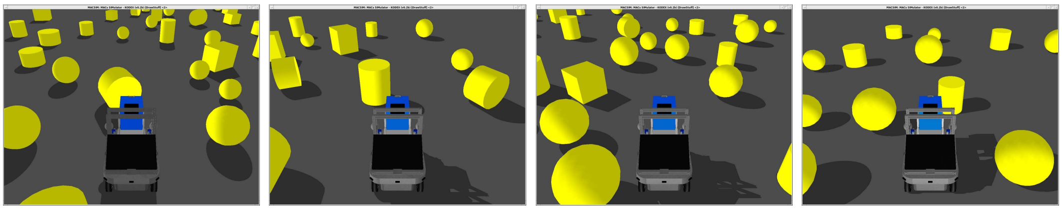
In summary, learning is conducted in an online-fashion, where first $n_{bootstrap}$ samples are collected for feature selection and initiating the classifier. Then the learning continues in a curiosity-driven way by selecting most interesting situations based on the distance threshold τ . As a result, two parameters, $n_{bootstrap}$ and τ determine the speed and performance of learning.

Learning is performed in MACSim, where the robot is placed in a $3{\times}3$ m^2 square room, which includes 100 randomly scattered objects with dimensions in the range $[20\text{c}m - 40\text{c}m]$. For each action, an online-SVM is trained using 3000 different samples, which are obtained by making 3000 different interactions in this room. During this phase, only the interesting samples are used in training the SVM (Fig. 4).

After training, the robot is transferred into another virtual room with similar characteristics and 2000 test samples are collected in the second room. These 2000 samples are used to evaluate and compare the performances of the controllers trained with differing values of $n_{bootstrap}$ and τ. In the next section we examine the effect of these two parameters on the speed and performance of the learning system, based on the system's prediction accuracy over the 2000 testing samples.

A. Effect of bootstrap period

The number of bootstrap samples, $n_{bootstrap}$ affects the quality of the feature selection process and the classifier's performance. If $n_{bootstrap}$ is large, the relevant features are more accurately selected, and more samples will be included in initial training without any curiosity check. In these experiments, in order to examine the effect of bootstrap period, the prediction accuracies of the classifier are computed for $n_{bootstrap}$ values of 10, 25, 50, and 100 on the testing set. In the box and whisker plot (Fig. 5), the prediction accuracy of the classifier on the test set is plotted against the bootstrap parameter, where each box represents the accuracy distribution of 10 different classifiers obtained from different orderings of the training samples. In this plot, for each value of the $n_{bootstrap}$, three successive boxes are drawn, corresponding to the prediction accuracy values at



(a) (b) (c) (d)

Fig. 4. These snapshots show example situations encountered in the learning phase. Curiosity-based learner found the two left-most situations interesting, executed go forward action and updated the the classifier based on the result of its actions. However the two right-most situations are found to be uninteresting and were not included in training. (a) Corresponds to a situation where boundaries of the cylinder's surface is similar to the sphere's from the robot's point of view, and the learner is required to be fine-tuned. (b) Corresponds to a situation where the object locates in the boundaries of the go-forward action. (c) The space in front of the robot is clear. (d) This situation seems to be similar to (b), however the (smaller) object in (d) is closer than the object in (b).

the 100$^{\text{th}}$, 250$^{\text{th}}$, and 400$^{\text{th}}$ interactions. When $n_{bootstrap}$ is selected as 10, the performance of the classifier remains below $\%90$ since 10 samples are insufficient for selecting the relevant features and bootstrapping an initial classifier with the ability to select interesting samples. The values greater than 25 does not further increase the performance, thus, 25 initial samples are found to be sufficient to bootstrap the learning process.

B. Effect of the curiosity parameter

The curiosity parameter τ determines the width of the band around the decision hyperplane of the SVM. As the τ gets larger, more samples will be selected as interesting. The effect of τ is examined by training different classifiers with different τ values (eg. 0.05, 0.10, 0.50, 1.00, and no curiosity). In the box and whisker plot (Fig. 6), the prediction accuracy of the classifier on the test set is plotted against τ, where each box represents the accuracy distribution of 10 different classifiers corresponding to different orderings of the training samples. Similar to the previous figure, for each value of the τ, three successive boxes are drawn, corresponding to the prediction accuracy values at the 100$^{\text{th}}$, 250$^{\text{th}}$, and 400$^{\text{th}}$ interactions. As shown, curiosity parameters that are too small keeps the system away from interacting with interesting situations. On the contrary, curiosity parameters that are too large slows down learning by including uninteresting samples in training. As a result, we selected $\tau = 0.50$ as the curiosity parameter to be used in the next section.

C. Using traversability affordance

In order to demonstrate the overall behavior of the robot, and its ability in perceiving the traversability affordance in the environment, it is placed in a room cluttered with objects of various shapes and size (Fig. 7). The controller used in this experiment was trained with $\tau = 0.5$ and $n_{bootstrap} = 50$. Here, the robot is additionally controlled by the motivation system which favors driving forward. Whenever the move-forward action is not afforded, a lower priority action is executed if it is afforded. As shown in the Fig. 7[1], the robot successfully wanders in the room. Note that the robot does not only drive towards the open-spaces, but if a higher priority action requires it, it chooses to drive over spherical and cylindrical objects in appropriate orientations, since they afford traversability. It also successfully avoids boxes and upright cylindrical objects by not driving towards them.

The controller used in the simulator is also transferred to the real Kurt3D robot. Various objects, including simple geometrical ones, and office environment object like trash bins and boxes are then placed on the way of Kurt3D to test the controller. As shown in Figure 8, the robot is able to correctly perceive the affordances of the box, cylindrical, and spherical objects, and act without colliding with non-

[1]Url:`http://www.kovan.ceng.metu.edu.tr/traversability/movie.mpg`

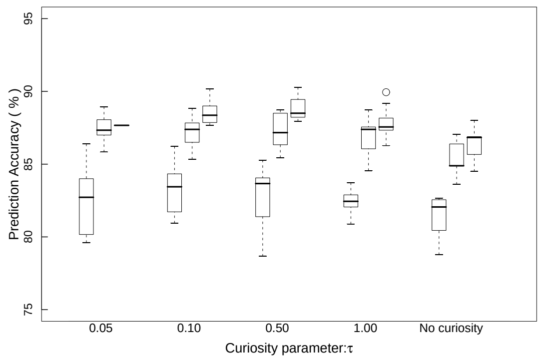


Fig. 6. The change in the prediction accuracy of the affordance perception during the learning phase. The thresholds which determine the curiosity level of the robot are compared. $n_{bootstrap}$ is fixed to 50.

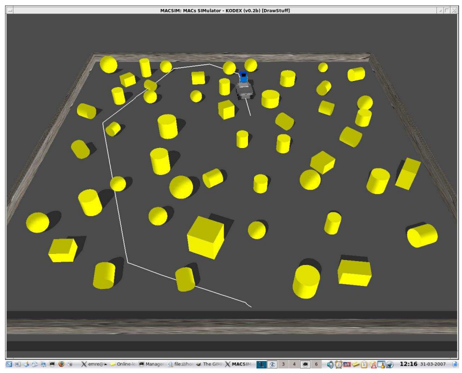


Fig. 7. The robot wanders in the room.

Fig. 8. The initial position of the robot is shown in the left-most figure. The robot first goes forward, then turns left since trash-bin does not afford traversability. Third snapshot shows the robot driving over the spherical object. The path of the robot is shown in the last figure.

traversable objects and driving over traversable ones.

V. CONCLUSIONS

In this paper, we studied the learning of traversability affordance on a mobile robot and investigated how the number of interactions required can be minimized with minimal degradation on the learning process. Specifically, we proposed a two step learning process which consists of bootstrapping and curiosity-based learning phases. In the bootstrapping phase, a small set of initial interaction data were used to find the relevant perceptual features for the affordance, and a Support Vector Machine (SVM) classifier was trained. In the curiosity-driven learning phase, a curiosity band around the decision hyperplane of the SVM was used to decide whether a given interaction opportunity is worth exploring or not.

The effects of two parameters of our learning system, τ and $n_{bootstrap}$, which serve as the curiosity threshold and number of bootstrap samples respectively, are examined in systematic experiments. Selecting τ small keeps the system away from interacting with interesting situations, and selecting it large slows down learning since uninteresting samples are used in training. As for $n_{bootstrap}$, while small values degrade the performance of the system, large values does not improve the performance after a certain threshold.

The affordance perception system, trained using optimized parameters, was tested in a room cluttered with objects of varying shapes. In this environment the robot was able to predict the traversability affordances of the objects, and wander around the room. The trained controller was also transferred to the real robot, which was also successful in predicting the traversability affordance of real world objects.

VI. ACKNOWLEDGMENTS

This work was partially funded by the European Commission under the MACS project (FP6-004381). URL: `http://macs-eu.org`

REFERENCES

[1] J. Gibson, *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Associates, 1986. Originally published in 1979.

[2] E. Gibson, "Perceptual learning in development: Some basic concepts," *Ecological Psychology*, vol. 12, no. 4, pp. 295–302, 2000.

[3] E. Gibson, "The world is so full of a number of things: On specification and perceptual learning," *Ecological Psychology*, vol. 15, no. 4, pp. 283–288, 2003.

[4] K. MacDorman, "Responding to affordances: Learning and projecting a sensorimotor mapping," in *IEEE Intl. Conf. on Robotics and Automation (ICRA 00)*, pp. 3253–3259, 2000.

[5] I. Cos-Aguilera, L. Canamero, and G. Hayes, "Motivation-driven learning of object affordances: First experiments using a simulated Khepera robot," in *In Proceedings of the 9th International Conference in Cognitive Modelling (ICCM'03)*, (Bamberg, Germany), April 2003.

[6] I. Cos-Aguilera, L. Canamero, and G. M. Hayes, "Using a SOFM to learn object affordances," in *In Proceedings of the 5th Workshop of Physical Agents*, (Girona, Catalonia, Spain), March 2004.

[7] P. Fitzpatrick, G. Metta, L. Natale, A. Rao, and G. Sandini, "Learning about objects through action -initial steps towards artificial cognition," in *IEEE Intl. Conf. on Robotics and Automation (ICRA 03)*, pp. 3140–3145, 2003.

[8] A. Stoytchev, "Toward learning the binding affordances of objects: A behavior-grounded approach," in *In Proceedings of AAAI Symposium on Developmental Robotics*, pp. 21–23, March 2005.

[9] A. Stoytchev, "Behavior-grounded representation of tool affordances," in *IEEE Intl. Conf. on Robotics and Automation (ICRA 05)*, (Barcelona, Spain), pp. 18–22, April 2005.

[10] G. Fritz, L. Paletta, M. Kumar, G. Dorffner, R. Breithaupt, and E. Rome, "Visual learning of affordance based cues.," in *Simulation of Adaptive Behavior (SAB)*, pp. 52–64, 2006.

[11] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner, "Intrinsic motivation systems for autonomous mental development," *IEEE Transactions on Evolutionary Computation*, 2007. to appear.

[12] E. Uğur, M. R. Doğar, O. Soysal, M. Çakmak, and E. Şahin, "MACSim: Physics-based simulation of the KURT3D robot platform for studying affordances," 2006. MACS Project Deliverable 1.2.1.

[13] D. Kim, J. Sun, S. M. Oh, J. M. Rehg, and A. Bobick, "Traversability classification using unsupervised on-line visual learning for outdoor robot navigation," in *IEEE Intl. Conf. on Robotics and Automation (ICRA 06)*, (Orlando, FL), May 2006.

[14] E. Ugur, M. R. Dogar, M. Cakmak, and E. Sahin, "The learning and use of traversability affordance using range images on a mobile robot," in *to appear in Proceedings of IEEE Intl. Conf. on Robotics and Automation (ICRA 07)*, April 2007.

[15] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *ML92: Proceedings of the ninth international workshop on Machine learning*, pp. 249–256, Morgan Kaufmann Publishers Inc., 1992.

[16] A. Bordes, S. Ertekin, J. Weston, and L. Bottou, "Fast kernel classifiers with online and active learning," *Journal of Machine Learning Research*, vol. 6, pp. 1579–1619, September 2005.