Autonomous Development of a Grounded Object Ontology by a Learning Robot

Joseph Modayil and Benjamin Kuipers
Department of Computer Sciences
The University of Texas at Austin

Abstract

We describe how a physical robot can learn about objects from its own autonomous experience in the continuous world. The robot identifies statistical regularities that allow it to represent a physical object with a cluster of sensations that violate a static world model, track that cluster over time, extract percepts from that cluster, form concepts from similar percepts, and learn reliable actions that can be applied to objects. We present a formalism for representing the ontology for objects and actions, a learning algorithm, and the results of an evaluation with a physical robot.

Introduction

We describe how a physical robot can learn about objects from its own autonomous experience in the continuous world. The robot develops an integrated system for tracking, perceiving, categorizing, and acting on objects. This is a key step in the larger agenda of developmental robotics, which aims to show how a robot can start with the "blooming, buzzing confusion" of low-level sensorimotor interaction, and can learn higher-level symbolic structures of commonsense knowledge. We assume here that the robot has already learned the basic structure of its sensorimotor system (Pierce & Kuipers 1997) and the ability to construct and use local maps of the static environment (Thrun, Burgard, & Fox 2005).

A learning robot developing knowledge about objects lies at the intersection of several research areas in AI, including autonomous robotics, machine learning, and knowledge representation. This work is answering part of the fundamental question of how human-level intelligence might arise in an autonomous learning robot exploring a complex continuous world. Starting with uninterpreted, continuous, high-dimensional sensorimotor experience, the learning robot generates knowledge representations that support symbolic inference and goal-driven behaviors. The learning process is made more challenging by the requirement that the robot must operate in real-time with limited resources (power, computation, and training data).

Learning about objects supports multiple research directions and is not merely an extension to robot map-building. Knowledge about objects has been used in several studies as the foundation for learning to communicate with language (Bloom 2001; Steedman 2002; Steels & Kaplan 2001; Yu, Ballard, & Aslin 2003; Roy & Pentland 2002). Object representations have also been used to apply symbolic inference and planning to physical robots (Hart, Grupen, & Jensen 2005). Algorithms that allow a robot to reason and interact with objects as readily as robots can represent and move through space will open up new domains for intelligent robots.

Objects and the actions they afford are intimately related, so in many ways their symbolic descriptions must be learned together. Objects are first separated from the background by identifying, grouping, and tracking elements of the sensory stream that are not adequately explained by the static background model. In many cases, these tracked objects have consistent shapes, which can be learned from the regularities in experience. A set of similar shapes forms a shape concept, and a learned concept can be used to generalize from past experience. Actions apply to objects, depend on context, and have reliable effects. The robot learns actions by using "motor babbling" to explore the space of contexts, motor signals, and effects, searching for extended control laws with relatively reliable effects. This learning process creates perceptual, structural, and functional representations of the objects. In experiments with a physical mobile robot, we demonstrate and evaluate this learning process.

In the following sections, we describe the formalism for representing the ontology for objects and actions, the algorithm for learning the ontology from experience, the evaluation with a physical robot, and related work.

Components of the Object Ontology

The ontology of objects is an abstraction of the low level continuous experience of the robot.

Continuous System

From an experimenter's perspective, a robot and its environment can be modeled as a dynamical system:

$$\begin{aligned} x_{t+1} &= F(x_t, u_t) \\ z_t &= G(x_t) \\ u_t &= H_i(z_0, \ldots, z_t) \end{aligned} \tag{1}$$

where x_t represents the robot's state vector at time t, z_t is the raw sense vector, and u_t is the motor vector. The functions

F and G represent relationships among the environment, the robot's physical state, and the information returned by its sensors, but these functions are not known to the robot itself.

The robot acts by selecting a control law H_i such that the dynamical system (Equation 1) moves the robot's state x closer to its goal, in the context of the current local environment. When this control law terminates, the robot selects a new control law H_j and continues onward.

The raw sensorimotor trace is a sequence of sense and motor vectors.

$$\langle z_0, u_0 \rangle, \langle z_1, u_1 \rangle, \cdots \langle z_t, u_t \rangle, \cdots \quad (2)$$

Symbolic Abstraction

We describe our object ontology by a tuple,

$$\langle Tra, Per, Con, Act \rangle \quad (3)$$

consisting of trackers (Tra), perceptual functions (Per), concepts (Con), and actions (Act).

An *object*, considered as part of the agent's knowledge representation, is a hypothesized entity that accounts for a spatio-temporally coherent cluster of sensory experience. Note that the word "object", when used in this sense, does not refer to a thing in the external world, but to something within the agent's knowledge representation that helps it make sense of its experiences.

A *tracker* $\tau \in Tra$ names two corresponding things:

1. the active process that tracks a cluster of sensory experience as it evolves over time, and
2. the symbol in the agent's knowledge representation that represents the object (i.e., the hypothesized entity that accounts for the tracked cluster).

A *perceptual function* f is used to generate the percept $f_t(\tau)$ which represents a property of τ at time t. The percept is formed from the sensory experience by the tracker τ. Examples of simple percepts include the distance, location, and color of a particular object at a particular time. A more complex percept is the shape of an object, which can be assembled from multiple observations over time.

For a particular perceptual function $f \in Per$, a *concept* $\sigma_f \in Con$ is an implicitly defined set of percepts similar to a prototype percept $q' = f'_t(\tau')$,

$$\sigma_f[q'] = \{q \mid d(q, q') \approx 0\}, \quad (4)$$

where d is a distance function (an example is given in Equation 13). For example, a shape concept is a set of shape percepts that are similar to a prototype shape percept. Figure 4 shows ten shape models, which are percepts obtained from the robot's sensory experience with the ten depicted objects. These individual percepts belong to ten concepts, each corresponding to percepts obtained from the same real-world object on different occasions.

An *action* $\alpha \in Act$ is specified by a description D of its effects on the object's percepts, the context C for the action to be reliable, and an associated control law H.

$$\alpha = \langle D, C, H \rangle \quad (5)$$

The next section describes how the robot can learn the components of this ontology.

Learning Object Representations

One goal of developmental robotics is for robots to be capable of learning both incrementally and without extrinsic rewards. Incremental acquisition allows the robot to learn from novel experience throughout its lifetime. The learned representations should also be generated through an autonomous, internal process. The following sections describe how the components of the ontology can be learned by a robot while satisfying these constraints from developmental robotics.

Formation of Trackers

Using the method from (Modayil & Kuipers 2004), a mobile robot can create trackers for movable objects. The robot senses the environment with a laser range finder. Each observation from the sensor is an array of distances to obstacles

$$z_t : \Theta \rightarrow R$$

as shown in Figure 1(a).

The robot uses these observations to construct an occupancy grid map of space as shown in Figure 1(b). The occupancy grid is constructed with the assumption that the world is static (Thrun, Burgard, & Fox 2005). In the occupancy grid, local space is divided into grid cells, each of which has some probability of being occupied or clear, and a SLAM algorithm updates these probabilities using sensor observations. In addition to this standard SLAM process, the algorithm marks each grid cell that is ever believed to be clear with high confidence.

When a physical object moves into a previously clear region in the map, the sensor readings that fall on the object violate the map's static world explanation. These readings are clustered spatially to define *snapshots*. A snapshot S of an object is a cluster of these dynamic range sensor readings in the map. An example of a snapshot is shown in Figure 1(b). Each snapshot is characterized by a circle that encompasses all the sensor readings.

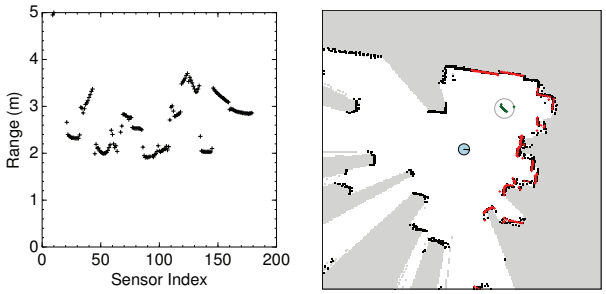
Finally, a tracker τ is created by forming associations between snapshots over time. The support of a tracker, $\text{Supp}_t(\tau)$, is given by the sensor indices of the points in the snapshots, and is represented as a subset of the sensor indices Θ. The tracker associates snapshots using their bounding circles. The tracker is terminated when clear successor snapshots do not exist.

Formation of Percepts

We define a small set of perceptual functions $f \in Per$ for the ontology. Each perceptual function gives rise to the percept $f_t(\tau)$ for a given tracker τ at a time t. The simplest percept is the object's support ($\text{Supp}_t(\tau)$). Localization in the occupancy grid is used to estimate the location and heading of the robot ($loc_t(\rho)$ and $head_t(\rho)$), which are non-object percepts.

Some percepts can be defined as functions of the object support. For this work we consider two such functions, but a larger set of functions could be generated autonomously using a constructive induction process (Shen 1990).

$$angle_t(\tau) = \text{mean}\{i \mid i \in \text{Supp}_t(\tau)\} \quad (6)$$



(a) (b)

Figure 1: (a) The sensor measures distances to obstacles, with readings taken at every degree. (b) By projecting the readings into an occupancy grid, the robot is able to identify readings from dynamic obstacles. A snapshot is formed by spatially clustering these readings, and forming a bounding circle.

$$dist_t(\tau) = \min\{z_t(i) \mid i \in \text{Supp}_t(\tau)\} \quad (7)$$

Another percept is the object's shape. A shape is represented with a set of situated views of the object, where each *situated view* is a tuple with the robot location, the robot heading, the object support and the sensor observation from a set of previous time steps I.

$$Shape_t(\tau) = \{\langle loc_{t'}(\rho), head_{t'}(\rho), \text{Supp}_{t'}(\tau), z_{t'}\rangle \mid t' \in I\} \quad (8)$$

When a tracker's shape matches a known shape (described in the next section), the robot is also able to generate percepts for the object's location and heading ($loc_t(\tau)$ and $head_t(\tau)$) in the map.

Formation of Concepts

Given a temporal sequence of percept values,

$$\ldots, f_{t-1}(\tau), f_t(\tau), f_{t+1}(\tau), \ldots$$

with a distance function d, a percept is a candidate for defining a concept if

$$\forall k > 0, \;\; d(f_t(\tau), f_{t+k}(\tau)) \approx 0. \quad (9)$$

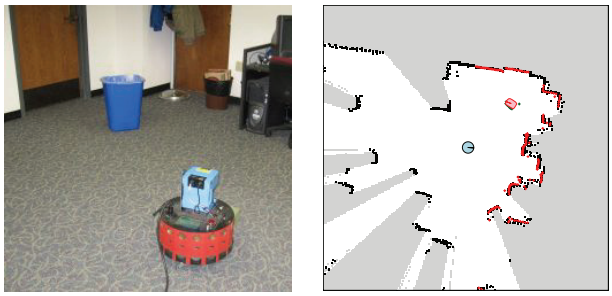
Thus, concepts are formed by creating clusters from object percepts that are stable in time. Concepts facilitate generalization from past experience.

Using a particular perceptual function $f \in Per$, a concept σ_f is defined by Equation 4 to be a set of percepts that are near the prototype percept $q' = f_{t'}(\tau')$ (within the threshold η).

$$\sigma_f[q'] = \{q \mid d(q, q') \leq \eta\} \quad (10)$$

The robot first checks an observed percept $f_t(\tau)$ to see if it is a member of a known concept. When the percept does not belong to a known concept, a new concept is generated from the percept.

We now describe how a concept is formed from a shape percept. First, structurally consistent shapes are created by minimizing violations of geometric constraints between the situated views in the shape percept. Figure 3 shows how error vectors can be defined between situated views. Using



(a) (b)

Figure 2: (a) A scene with the learning robot observing a physical object. (b) The robot builds a structured description of its local environment consisting of a static map, the learning robot, and the recognized object.

three error vectors defined in the figure, we define an inconsistency measure for an object shape A,

$$\mu(A) = \sum_{a \in A} \sum_{b \in A} ||e_{L,a,b}||^2 + ||e_{R,a,b}||^2 + ||e_{I,a,b}||^2. \quad (11)$$

Minimizing this error generates consistent shapes, as shown in Figure 4. The minimization is performed by numerical optimization.

We denote the rigid transformation of a shape B (defined as a set of tuples in Equation 8) by offset vectors for the location and heading (λ and γ respectively) by

$$T_{\lambda,\gamma}(B) = \{\langle l + \lambda, h + \gamma, S, z\rangle \mid \langle l, h, S, z\rangle \in B\}. \quad (12)$$

The distance between two shapes is then defined to be the minimum error over all rigid transformations.

$$d(A, B) = \min_{\lambda,\gamma} \mu(A \cup T_{\lambda,\gamma}(B)) \quad (13)$$

This distance function is used by the robot to create the shape concepts shown in Figure 3 (with $\eta = .02$ in Equation 10).

New percepts for the object location and heading are also defined using this distance function. Given an object shape A, the robot estimates its pose in the object frame of reference, by searching for a robot location and heading that are consistent with the tracker τ,

$$\lambda', \gamma' = \arg\min_{\lambda,\gamma} d(A, \{\langle \lambda, \gamma, \text{Supp}_t(\tau), z_t\rangle\}). \quad (14)$$

Once the robot knows its location and heading in both the reference frame of the map and the reference frame of the object, the robot can estimate the location $loc_t(\tau)$ and orientation $head_t(\tau)$ of the object in the map.

Formation of Actions

Thus far, the robot does not have mechanisms for interacting with the object. To address this need, we now describe how the robot learns actions that reliably change individual object properties. Our definition of an action differs from STRIPS actions (with complete declarative preconditions and post-conditions), and reinforcement learning actions (with no

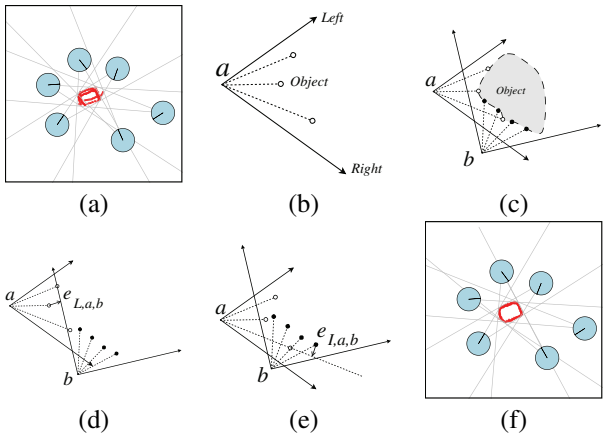


Figure 3: (a) The shape percept is a set of situated views of the physical object. A single situated view consists of the robot's pose, the tracker's support and the sensory observation. (b) The object is bounded by rays on the left and right. (c) The sensor readings from one situated view must fall within the bounding rays from all other situated views. (d) Exterior error vectors are defined from violations of this geometric constraint. A left error vector ($e_{L,a,b}$) is shown here and a right error vector ($e_{R,a,b}$) is defined similarly (Modayil & Kuipers 2006) (e) An interior error vector ($e_{I,a,b}$) is defined from sensor readings that come from the inside of an object. (f) A consistent shape description is created by minimizing the lengths of these error vectors.

declarative description). Our definition of an action (Equation 5) possesses a partial description of an action's postconditions, along with complete declarative preconditions. The partial description of effects simplifies learning, but it can limit the reliability of plans.

Actions facilitate planning by characterizing the behavior of a control law. To sequence actions, the planner must know the preconditions of a control law and its effects. We form actions by learning control laws whose effects and preconditions have simple descriptions.

The robot learns actions by observing the effects of performing random motor babbling in the presence of the object. Motor babbling is a process of repeatedly performing a random motor command for a short duration. One strength of the action learning approach we present here is that the robot is able to use this goal-free experience to form actions that can be used for goal directed planning. The robot performs self-supervised learning, where the observations in the training data are labeled using the qualitative changes that occur to an individual percept. The learned actions can be used to achieve goals by reducing the difference between the robot's current perception and the desired goal, as demonstrated in the evaluation. Thus goals are not required while the actions are being learned but they are required for planning and execution.

Action Definition An action is defined in Equation 5 as a tuple with a description, a context and a control law. These components are now formally defined.

Figure 4: A set of physical objects with their learned shapes.

Our action learning algorithm is restricted to perceptual functions f_j which are either vector-valued or not perceived.

$$f_{j,t}(\tau) \in \Re^{n_j} \cup \{\perp\}. \tag{15}$$

In particular, the robot does not learn an action to change the shape of an object, as a shape percept represented by a set. The change in this perceptual function is denoted by δ,

$$\delta_{j,t} = f_{j,t+1} - f_{j,t}. \tag{16}$$

The description of an action,

$$D = \langle j, b, q_b \rangle,$$

consists of the name j of the perceptual function to be controlled, the qualitative behavior b,

$$b \in \{up, down\} \cup \{dir[f_k] \mid f_k \in Per\}, \tag{17}$$

and the quantitative effect q_b. Two qualitative behaviors are defined for a scalar perceptual function: going up and going down. The qualitative behavior $dir[f_k]$ for a vector function f_j means that f_j changes in the direction of f_k. The quantitative effect for scalars is bounded by ϵ,

$$q_{up}(\delta_t) \equiv \delta_t > \epsilon, \quad q_{down}(\delta_t) \equiv -\delta_t > \epsilon.$$

The quantitative effect for vectors is bounded by ϵ and ϵ',

$$q_{dir[f_k]}(\delta_t) \equiv ||\delta_t|| > \epsilon \wedge \frac{\langle \delta_t, f_{k,t} \rangle}{||\delta_t|| \cdot ||f_{k,t}||} > 1 - \epsilon'.$$

The context of an action is represented as a conjunction of inequality constraints on scalar perceptual functions:

$$(xRc) \text{ where } x \in \{f_k\},\ R \in \{\leq, \geq\},\ c \in \Re. \tag{18}$$

Finally, the control law of an action is a function H from a percept to a motor output. In this work we restrict our attention to constant functions.

Learning Algorithm Learning an action that satisfies a qualitative description amounts to defining the components of the action as defined in Equation 5. First, to complete the description, a threshold ϵ is selected from the observed values of δ. Next, the quantitative effect is used to search for constraints on the perceptual context and motor output that reliably induce the desired behavior. Finally, the constraints are used to define a perceptual context and a control law.

For each perceptual function, a threshold ϵ is chosen by running a Parzen window with a Gaussian kernel over the observations of δ (or $||\delta||$ for vector percepts). The threshold ϵ is set to the first local minimum above zero if it exists, otherwise it is set to a value one standard deviation from the mean. The value of ϵ' for vector percepts is set by optimization with the context in the utility defined below.

The threshold ϵ is used to define q_b, and q_b is used to label the examples in the training data. The learning algorithm uses the labeled examples to search for constraints on the percepts and motor outputs that generate the desired behavior. The constraints are represented by axis aligned half-spaces, specified as inequalities over the variables of the scalar perceptual functions (f_k) and the components of the motor vector ($\pi_k(u)$).

To find the perceptual constraints C and motor constraints M, we define a set of measures for the precision (μ_0), recall (μ_1), and repeatability (μ_2). The utility function U is their geometric mean. These functions are defined using the empirical probability (Pr) as measured in the training data.

$$\begin{aligned} \mu_0 &= Pr(q_b(\delta_t) \mid z_t \in C \wedge u_t \in M) \\ \mu_1 &= Pr(z_t \in C \wedge u_t \in M \mid q_b(\delta_t)) \\ \mu_2 &= Pr(z_{t+1} \in C \mid z_t \in C \wedge u_t \in M) \\ U &= (\mu_0 \mu_1 \mu_2)^{\frac{1}{3}} \end{aligned} \tag{19}$$

Constraints are added incrementally to greedily optimize the utility function. The process terminates when adding a constraint provides no significant improvement to utility. The newly generated action is discarded if the final utility measure is low. Otherwise, the learned context C becomes part of the action.

A constant control law is defined from M.

$$H(z_t) = m = \arg\min_{u \in M} ||u|| \tag{20}$$

The constant control law is enhanced in two ways. The first is to account for perceptual latencies by predicting the current value of the percept. The second is to scale the motor output by the minimum effect ϵ, when the robot wants to change a percept to a goal value g.

$$s(g, \tau) = \min(1, ||E[f_{k,t}(\tau)] - g||/\epsilon) \tag{21}$$

$$H(z_t) = s(g, \tau) \cdot m \tag{22}$$

Putting the learned components together creates the new action $\alpha = \langle D, C, H \rangle$.

Training Scenario The above algorithm was used to learn actions. The robot first gathered observations by randomly selecting a motor command and executing it for a fixed duration. The motor commands for drive and turn (linear and angular velocities) were selected from the following set.

$$\{-0.2, 0.0, 0.2\} m/s \times \{-0.4, 0.0, 0.4\} rad/s$$

Action I	Action II	Action III
Description :	**Description** :	**Description** :
angle(τ)	dist(τ)	loc(τ)
up	down	dir[head(ρ)]
$\delta > 12$	-$\delta > .19$	$\|\|\delta\|\| > .21$, $\epsilon' = .13$
Context:	**Context**:	**Context**:
$\emptyset$	dist ≥ 0.43	dist ≤ 0.22
	angle ≤ 132	angle ≥ 77
	angle ≥ 69	angle ≤ 112
Control Law:	**Control Law**:	**Control Law**:
drive = 0.0 m/s	drive = 0.2 m/s	drive = 0.2 m/s
turn = -0.4 rad/s	turn = 0.0 rad/s	turn = 0.0 rad/s

Figure 5: The above actions were learned by the robot from its observations of the effects of motor babbling. These actions cause changes in (I) the angle to the object (by turning), (II) the distance to the object (by driving), and (III) the location of the object in the map (by pushing).

The data was gathered in different environmental configurations, where the experimenter changed the environment between trials. Running the above algorithm generated several useful actions, examples of which are shown in Figure 5. These actions may be thought of as simple affordances of the object, which the robot currently assumes will always work. However, the action for pushing an object could fail for heavy objects. In the future, the robot could potentially create new perceptual functions to predict which objects are not pushable.

Representing and Achieving Goals Part of the value of the learned representations is that it provides a language for representing goals. The high-level task given by "Place a recycle bin in the center of the room" can be represented as a goal state with a tracker whose shape corresponds to a recycle bin and whose location is in the center of the room. The robot can set goals and measure its progress towards achieving them. The learned actions are used by a planner to achieve goals by sequentially reducing differences between the robot's current percepts and the goal.

To achieve goals, the constraints provided in an action's context are used with backchaining to create reactive plans to change a percept to a goal value. Attempting to satisfy the preconditions sequentially can fail when more than one precondition is not satisfied. In this situation, the robot simulates observations from multiple poses to find a pose from which the perceptual preconditions are satisfied. The robot moves to this pose and then executes the desired action.

Evaluation

We have described algorithms that generate object representations for robots. Now we demonstrate the utility of these representations on our mobile robot. Our experimental platform is a Magellan Pro robot with a laser rangefinder running with Player drivers (Gerkey, Vaughan, & Howard 2003). The laser rangefinder provides a planar perspective of the world, from approximately 30cm above the ground. We evaluated the learned ontology on two tasks. The first is a classification task that tests the robot's object recogni-

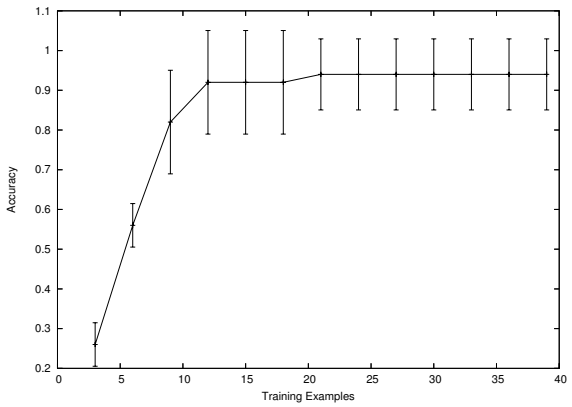


Figure 6: The graph of classification accuracy shows that the robot's internal object concepts correspond well with the physical objects. The error bars show standard deviations.

Behavior	Distance	Accuracy	Time (s)
Face	45 °	7.4° ($\sigma=3$)	4.38 ($\sigma=.51$)
Approach	1.8 m	.04 m ($\sigma=.02$)	21.1 ($\sigma=5.7$)
Move	2.0 m	.09 m ($\sigma=.04$)	258 ($\sigma=138$)

Table 1: The robot used the learned actions to perform three tasks: facing the object, approaching the object and moving the object. The columns indicate the initial distance to the goal, the final distance from the goal and elapsed time. Ten runs were performed for each task, and the results are shown with the standard deviations. All trials succeeded with the robot accurately achieving its goals. The time to task completion has a high variance since the robot keeps trying until it succeeds.

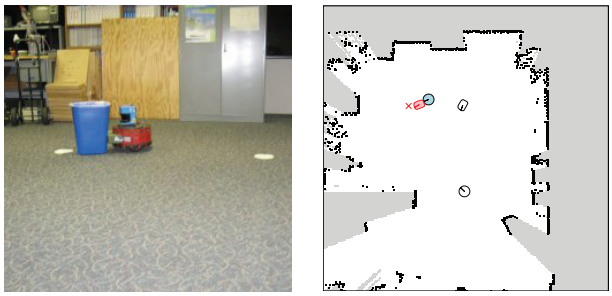
tion capability. The second task tests the robot's ability to achieve goals.

Classification

We evaluated the robot's ability to perform an object classification task. The robot modeled the shape of each of the ten objects in Figure 4 on five separate occasions. The classification task is to predict the object's true identity as provided by the experimenter. The robot performed instance based learning, associating a label to a concept from its defining training example. The results from a five fold stratified cross validation experiment are shown in Figure 6. The results show that supervised learning using the autonomously learned shape concepts is effective for this task. A perfect learner would achieve 100 percent accuracy after ten examples (one example for each class), while random guessing would only achieve 10 percent accuracy.

Interaction Tasks

To evaluate the learned actions, we measured the ability of the robot to perform three tasks: facing the object, approaching the object and moving the object to a location. These tasks were represented by setting goal values for the $angle(\tau)$, $dist(\tau)$ and $loc(\tau)$ percepts respectively. The starting state for the three tasks was approximately the same (the object was placed at different orientations), and is shown in Figure 7. The desired final states for the tasks were to have the object in front of the robot ($|angle-90| < 10$), to have the robot near the object ($dist \leq 1.0$), and to have the object at the goal location in the figure ($||loc_t(\tau)-(3,2)|| <$ .15 m). Ten runs were performed for each task, and the experimenter physically verified task completion for each run. The results in Table 1 show that the robot is able to achieve these goals reliably and accurately. Figure 7 shows an example of the robot pushing the object to a goal.



(a) (b)

Figure 7: (a) The robot pushes a recycling bin towards a goal location. (b) The shaded shapes show the robot's percepts for itself and the object. The starting poses of the robot and the object are shown unshaded, and the goal location for the object is indicated by $\times$.

Related Work

Work in psychology has explored the development of object representations in children. Work by Spelke (1990) has studied how children develop from using motion as a indicator of object unity to using other cues. Work by Mandler (2004) has explored how concepts might form in more general conditions. Work by Bloom (2001) has studied how objects and concepts are used to quickly learn a language.

Previous work in developmental robotics (Pierce & Kuipers 1997; Philipona, O'Regan, & Nadal 2003; Choe & Smith 2006) has shown how the structure of an agent's sensory and motor systems can be learned. A key method in this process is the projection of high-dimensional observations into a low-dimensional space. Further advances include Isomap (Tenenbaum, de Silva, & Langford 2000) which identifies manifolds in the data, and the use of information distance (Olsson, Nehaniv, & Polani 2006).

There is less work studying how a robot can learn object representations with actions. Work by Stoytchev (2005) has a robot learning the affordances of simple tools. Other work (Hart, Grupen, & Jensen 2005) demonstrates how a robot can learn the preconditions for actions. Natale (2004) has shown how motor babbling with a robot arm can be used to learn how to move objects. The work in these papers use stationary perception, whereas our work demonstrates the use of mobile perception.

Related work has also explored object recognition and action but not in conjunction. Work on object recognition has used mapping techniques (Biswas *et al.* 2002), and image-based models (Li, Fergus, & Perona 2003). Actions have

been learned in simulated symbolic domains (Benson 1995; Zettlemoyer, Pasula, & Kaelbling 2005). These approaches provide methods for learning actions when an object model is already available, but do not address how continuous actions can be learned on a mobile robot.

Discussion and Future Work

The above work shows how a robot starting with an understanding of space can construct object representations. Multiple representations are acquired, which can be interpreted as perceptual, structural and functional models. The object snapshots and trackers form perceptual representations. The shape model provides a structural representation. A functional model is available through the learned action of pushing: this action could be used to define the pushing affordance of an object. By integrating these different aspects of objects, the learned representations support perception, geometric inference, and goal-directed planning.

These representations are part of an ontology of objects grounded in the robot's sensorimotor experience. The learned ontology creates object trackers of individual objects, forms percepts from observations, forms concepts to generalize from past experience, and learns actions to change the perceptual properties of an object. Using this ontology, the physical robot is able to recognize objects and plan with learned actions to achieve goals. The learned ontology is simple and lays the foundation for learning more complex object models. In future work, the learned object representations may be used to extend a robot's ability to understand and interact with its environment.

Acknowledgments

The authors would like to thank Patrick Beeson and Aniket Murarka for their insightful comments. This work has taken place in the Intelligent Robotics Lab at the Artificial Intelligence Laboratory, The University of Texas at Austin. Research of the Intelligent Robotics lab is supported in part by grants from the National Science Foundation (IIS-0413257 and IIS-0538927), from the National Institutes of Health (EY016089), and by an IBM Faculty Research Award.

References

Benson, S. 1995. Inductive learning of reactive action models. In *Int. Conf. on Machine Learning*, 47–54.

Biswas, R.; Limketkai, B.; Sanner, S.; and Thrun, S. 2002. Towards object mapping in non-stationary environments with mobile robots. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1014–1019.

Bloom, P. 2001. Précis of How Children Learn the Meanings of Words. *The Behavioral and Brain Sciences* 24(6):1095–1103.

Choe, Y., and Smith, N. H. 2006. Motion-based autonomous grounding: Inferring external world properties from encoded internal sensory states alone. In *Proc. Nat. Conf. Artificial Intelligence (AAAI-2006)*.

Gerkey, B.; Vaughan, R. T.; and Howard, A. 2003. The Player/Stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11th Int. Conf. on Advanced Robotics*, 317–323.

Hart, S.; Grupen, R.; and Jensen, D. 2005. A relational representation for procedural task knowledge. In *Proc. 20th Nat. Conf. on Artificial Intelligence (AAAI-2005)*.

Li, F.-F.; Fergus, R.; and Perona, P. 2003. A Bayesian approach to unsupervised one-shot learning of object categories. In *Proc. IEEE Conf. on Comp. Vision*, 1134–1141.

Mandler, J. 2004. *The Foundations of Mind: Origins of Conceptual Thought*. Oxford University Press.

Modayil, J., and Kuipers, B. 2004. Bootstrap learning for object discovery. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 742–747.

Modayil, J., and Kuipers, B. 2006. Autonomous shape model learning for object localization and recognition. In *IEEE Int. Conf. on Robotics and Automation*, 2991–2996.

Natale, L. 2004. *Linking Action to Perception in a Humanoid Robot: A Developmental Approach to Grasping*. Ph.D. Dissertation, LIRA-Lab, University of Genoa, Italy.

Olsson, L.; Nehaniv, C.; and Polani, D. 2006. From unknown sensors and actuators to actions grounded in sensorimotor perceptions. *Connection Science* 18(2):121–144.

Philipona, D.; O'Regan, J. K.; and Nadal, J.-P. 2003. Is there something out there? Inferring space from sensorimotor dependencies. *Neural Computation* 15:2029–2049.

Pierce, D. M., and Kuipers, B. J. 1997. Map learning with uninterpreted sensors and effectors. *Artificial Intelligence* 92:169–227.

Roy, D. K., and Pentland, A. P. 2002. Learning words from sights and sounds: a computational model. *Cognitive Science* 26:113–146.

Shen, W.-M. 1990. Functional transformations in AI discovery systems. *Artificial Intelligence* 41(3):257–272.

Spelke, E. S. 1990. Principles of object perception. *Cognitive Science* 14:29–56.

Steedman, M. 2002. Formalizing affordance. In *Proceedings of the 24th Annual Meeting of the Cognitive Science Society*, 834–839.

Steels, L., and Kaplan, F. 2001. Aibo's first words : The social learning of language and meaning. *Evolution of Communication* 4(1).

Stoytchev, A. 2005. Behavior-grounded representation of tool affordances. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 3071–3076.

Tenenbaum, J. B.; de Silva, V.; and Langford, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science* 290:2319–2323.

Thrun, S.; Burgard, W.; and Fox, D. 2005. *Probabilistic Robotics*. Cambridge, MA: MIT Press.

Yu, C.; Ballard, D. H.; and Aslin, R. N. 2003. The role of embodied intention in early lexical acquisition. In *Proc. 25th Annual Meeting of the Cognitive Science Society*.

Zettlemoyer, L.; Pasula, H.; and Kaelbling, L. P. 2005. Learning planning rules in noisy stochastic worlds. In *Proc. 20th Nat. Conf. on Artificial Intelligence (AAAI-2005)*, 911–918.