Learning to Reach by Building a Representation of Peri-Personal Space

Jonathan Juett[1] and Benjamin Kuipers[1]

***Abstract*— Inspired by the early spatial learning of human infants, we describe progress toward enabling a robotic learning agent to learn the structure of *peri-personal space* — the space immediately around the agent within which reaching and grasping take place — with minimal prior spatial knowledge.**

We propose the *PPS Graph* representation for early knowledge of peri-personal space, a model that produces behaviors qualitatively consistent with early human motion, and that may provide information on how humans learn manipulation skills through its implementation in a computational system. Each graph node represents a visual sense vector and a proprioceptive sense vector corresponding to the same state of the world, but neither sense vector has a pre-existing interpretation in terms of a 3D model of the environment. An edge linking two nodes in the PPS Graph represents the feasibility of motion between those two states.

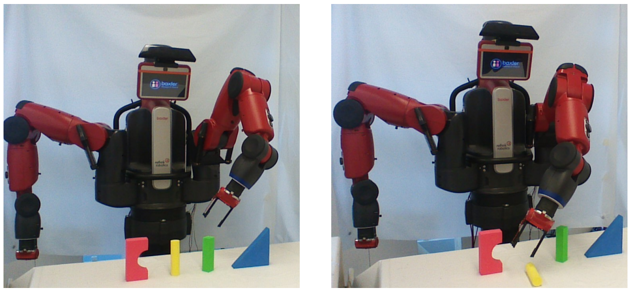
Learning starts with "motor babbling", random exploration of the space of joint angle vectors, leading to the creation of an initial PPS graph representing arm configurations in an otherwise empty space. The next crucial step is recognizing an unusual event, such as accidentally colliding with an object and changing its position. Once a type of unusual event has been identified, the goal for learning is to identify the prerequisites for an action to achieve an event of that type.

We report the results of experiments on a physical Baxter robot, both on a small Learning Graph and on a much larger Sampled PPS Graph to demonstrate scalability. We show how appropriate features can be extracted from uninterpreted visual images, and that combining weakly informative features with Naïve Bayes allows our robot to plan and make reliable reaching motions. We hypothesize that a similar approach will extend these results to grasping and moving objects.

I. INTRODUCTION

Peri-Personal Space (PPS) is defined as the space immediately around the agent that it can affect with its manipulators. The agent can sense this space initially via vision and proprioception, and later, touch. The problem we address is how an agent can learn from its own unguided experience to represent and use the structure of peri-personal space.

Experts in computer vision [9] and robot manipulation [15] have developed powerful methods for using precise prior knowledge of the physical properties of a robot to reconstruct the geometry of the environment and its manipulators. Although these methods are useful, they shed little light on how an intelligent agent with minimal externally provided knowledge can acquire these skills. By contrast, a newborn human baby starts with random undirected motions, and learns within a few months to interact purposefully with objects in its peri-personal space.



(a) (b)

Fig. 1. (a) Our physical Baxter Research Robot, shown with foreground objects within its peri-personal space. (b) Our model allows the agent to plan and execute motions that cause persistent changes in a target object (the yellow cylinder in this case), while avoiding the other objects. Through a series of experiments, we demonstrate that the agent learns visual prerequisites for reliable actions for reaching a target object. The endpoint of a reach is planned by using a graph representation of peri-personal space to map from the desired visual appearance to the configuration that produced it. Reaches planned in this way successfully Bump the target object in 90% of trials, demonstrating the success of our model for autonomous learning of early manipulation skills with minimal assumptions.

A. Constraints from Developmental Psychology

The infant exhibits an impressive ability to adapt to its changing body and sensory and motor capabilities. We look to developmental research with human infants for clues about effective representations for knowledge and effective learning methods. Fortunately, learning to reach has been a focus of developmental research for many decades [3], [7].

From birth to about 15 weeks, infants can respond to visually perceived objects, but during this "pre-reaching" phase, they cannot reliably achieve contact [3]. From pre-reaching until about eight months of age, reaching movements become increasingly successful, but remain jerky, with successive submovements, some of which may represent corrective submovements [19], and some of which reflect underdamped oscillations on the way to an equilibrium point [17].

Since Piaget [12] and before, it was generally believed that reaching was visually guided by feedback about the relation between the hand and the target object. However, it has since become clear that [7, p.1], "from their earliest attempts, infants can reach in the dark toward a glowing target without seeing their hand [6]."

After eight months, motion becomes less jerky, a single primary movement dominates the reach as it does in older children and adults, and vision of the hand becomes increasingly important for shaping the hand in anticipation of contact and grasping the target object [3]. Many aspects of

[1] Division of Computer Science and Engineering, University of Michigan, 2260 Hayward St, Ann Arbor, MI 48109. {jonjuett, kuipers} at umich.edu.

the evolution of reaching motion control can be explained by dynamical systems theory [17] and by optimal control and reinforcement learning [4].

This raises a key question [7, p.2]: "*What remains unclear is how looking at the object and bringing the hand to that location occurs at first when infants perform their initial intentional attempts to hit the target. What visuo-motor mapping process allows this to happen?*"

B. Robotic Models of Learning to Reach

Savastano and Nolfi [14] is the most comparable embodied computational model of learning to reach and grasp. Using a simulated model of the iCub robot, they demonstrate pre-reaching, gross-reaching, and fine-reaching phases of learning and behavior. They describe their results as qualitatively matching observations of children such as diminished use of vision in the first two phases, and proximal-then-distal use of the arm's degrees of freedom.

Their learning mechanism is a recurrent neural network, and the transitions from one phase to the next are represented by adding certain links and changing certain parameters in the network. This does result in qualitative changes in the trajectories of the robot's hand, but it begs the question about how and why those changes take place.

Chinellato, et al, [5] describe an approach to mapping peri-personal space based in the properties of certain brain regions. They assume that all spatial regions are represented in terms of Euclidean frames of reference, so peri-personal space is learned by learning the parameters of built-in frames of reference for eye, head, body, and arm.

Ugur, et al [18] demonstrate autonomous learning of behavioral primitives and object affordances, and imitation learning of complex actions. However, they start by assuming that peri-personal space can be modeled as a 3D Euclidean space, and that hand motions can be specified via starting, midpoint, and endpoint coordinates. Our agent does not assume a prior representation for peri-personal space, instead building the abstract PPS Graph representation without spatial assumptions and from uninformed experience.

C. Our Representation for Peri-Personal Space

We present initial results showing how the agent can learn its own model of peri-personal space. In our hypothesized model, a physically-embodied robot learning agent starts with minimal knowledge of its sensors, its effectors, and its environment. It learns a graph (the PPS Graph) which represents a correspondence between visual and proprioceptive perception of the environment, including its own body.

Each node n_i in the PPS Graph represents a state of the arm in the environment, and is associated with the visual sense vector $p_i = P(n_i)$ and the proprioceptive sense vector $q_i = Q(n_i)$ observed at that state. The visual sense vector $P(n_i) = \langle P(n_i, 1), \ldots, P(n_i, C)\rangle$ is the vector of images $P(n_i, c)$ the robot receives from its cameras $c \in \{1, \ldots, C\}$. In our case, there are three fixed cameras ($C = 3$) and each $P(n_i, c)$ is a low-resolution (160×120) full-color 2D image. Visual systems such as mobile cameras mounted on the robot or an RGB-D camera should also allow learning of analogous features and behaviors, but are not tested in this work. The proprioceptive sense vector $q_i = Q(n_i)$ is the vector of joint angles of the arm. We define the distance between nodes n_i and n_j as the Euclidean distance between q_i and q_j.

While creating the graph, the agent also makes visual observations of the motion along graph edges by taking videos with each camera for the duration of the motion. $V(e_{ij}, c)$ is a video taken by camera c as the robot moves from n_i to n_j. The frames of this video $V(e_{ij}, c)(t)$ are images indexed by frame $t \in [0, length(V(e_{ij}, c))]$. $V(e_{ij})$ denotes the vector of videos, indexed by c.

We assume that the agent is able to use simple image processing techniques to generate binary images corresponding to important segments of images. Given an image I_c of the form $P(n_i, c)$ or $V(e_{ij}, c)(t)$, $R(I_c, c)$ is a binary image, specifying the segment of I_c corresponding to the robot. Likewise, $G(I_c, c) \subseteq R(I_c, c)$ is the gripper segment in I_c. $R(I)$ and $G(I)$ denote vectors of binary images corresponding to the vector of images I, indexed by c.

The agent is able to identify the region of the view of camera c that the gripper sweeps through in its motion along a given e_{ij} by processing $V(e_{ij}, c)$ to find

$$G(e_{ij}, c) = \cup_t \left[G(V(e_{ij}, c)(t))\right], \tag{1}$$

which is a binary image that corresponds to all pixels that displayed the gripper in at least one frame of $V(e_{ij}, c)$. $G(e_{ij})$ refers to the vector of such binary images, indexed by c. Note that $G(e_{ji})$ is not necessarily equal to $G(e_{ij})$, so e_{ij} and e_{ji} must be represented as separate directed edges, and not as a single undirected edge. If $V(e_{ij}, c)$ does not exist for some e_{ij}, $G(e_{ij}, c)$ can be approximated by the convex hull of $G(n_i, c) \cup G(n_j, c)$.

D. Properties of the PPS Graph Representation

In the *pre-reaching* phase, the PPS graph is learned by defining nodes for the simultaneous visual and proprioceptive input sense vectors for a collection of states, visited during exploration including "motor babbling." The initial and final states of each motion are represented as nodes in the PPS graph. A directed edge e_{ij} connecting two nodes n_i and n_j represents a feasible motion between them, and is associated with the vector of binary images $G(e_{ij})$ representing the space swept by the gripper (eqn. (1)).

When an object is seen in peri-personal space, edges are retrieved from the PPS graph where $G(e_{ij})$ overlaps with the segment corresponding to that object in each of the current camera images. These edges and their endpoint nodes provide the proprioceptive vectors associated with states of the arm most likely to interact with that object. A path in the PPS graph, starting at the current arm pose and ending by sweeping through one of these matching edges, defines a trajectory for moving the arm to reach that object.

a) Generality: Our knowledge representation is designed to make weak assumptions about the nature of the sensors, effectors, and environment, in order to be general across significantly different sensory and motor systems, human and robotic. Specifically, we represent peri-personal space with a graph rather than a 3D Euclidean space.

To demonstrate this generality, we implemented a version of our framework with a multi-monocular visual perceptual system. Multiple cameras are preferred to a single camera to better distinguish occlusions from true intersections, avoiding overly conservative motion around obstacles and poor accuracy in reaching targets. The system is *not* given the relative poses of the cameras, and does *not* use the images to reconstruct the 3D structure of the scene.

We assume that the perceptual system can discriminate between background and foreground elements of images. During the pre-reaching phase, the background is initially static and neutral, with the arm being the only foreground object. Later, distinctively colored objects are placed in the environment, and are perceived as part of the foreground.

b) Focusing on Unusual Events: In our model, learning the map of peri-personal space is driven by the detection of *unusual events*, and the process of learning reliable preconditions for the actions that cause those unusual events.

During the pre-reaching phase, the only recognizable change is the motion of the arm. Everything else in the scene remains static, and thus is reliably predictable. However, it will sometimes happen that the arm (often the gripper) will contact an object, and the vision system will recognize an unusual event: a previously static object has moved.

On defining the event of moving an object, the learning agent is driven by *intrinsic motivation* [1] to learn to predict reliably when this event will take place, and thus to be able to cause the event to happen. This motivates a reinforcement learning process to learn the prerequisites for acting to cause this event. We call this action a *reach*, and this is the *gross reaching* phase of learning.

During the gross reaching phase, another unusual event is to *grasp* an object, which means to temporarily bind the object to the hand, so the object moves along with the hand.

The *palmar reflex* [8] makes this grasping action merely unusual, rather than astronomically rare, so the agent can experience enough examples to support learning of prerequisites. The phase where the prerequisites for the grasp action are being learned is called the *fine reaching* phase. (This paper focuses on the pre-reaching and gross reaching phases, and leaves the fine reaching phase for future work.)

c) Learning Prerequisites for Unusual Events: To detect the motion of an object, we need perceptual features that can be defined over the raw pixels of the vector of images. We assume the ability to individuate different objects, and the ability to measure the overlap between images of different objects, or images of the same object at different times, by computing the *intersection over union* (IOU) feature:

$$IOU(A, B) = |A \cap B|/|A \cup B| \tag{2}$$

where A and B are the sets of pixels in the images of the two objects, and $|S|$ is the number of pixels for a set S.

If object A is unchanged from time t to t', we expect that $IOU(A(t), A(t')) \approx 1$. Likewise, if $IOU(A(t), A(t')) \ll 1$ we recognize that a change has taken place.

II. EXPERIMENTS

We generate sets of potential features and learn which ones accurately predict the change in the object. Over the course of the experiments presented in this paper, the agent learns to use these predictive features to generate reliable actions that allow purposeful repetition of the same type of change. Our goal is to demonstrate that the agent gains an understanding of its peri-personal space by building and using the PPS Graph. We evaluate this goal indirectly by testing the ability of the agent to perform reaching and avoidance tasks. Since maximizing success rates is not our primary focus, we choose not to compare our agent's performance against that of state-of-the-art, information-rich methods. Similarly, we do not attempt to solve the tasks with a motion planner or additional data, as this would weaken the parallels between our approach and the development of peri-personal space understanding and manipulation skills in humans.

A. Experiment 1 - Sampling to Build a Dense PPS Graph

We first determine whether a graph of reasonable size provides sufficiently dense coverage of peri-personal space. The goal is to obtain a dense set of nodes that is a representative sampling of the feasible points in the agent's configuration space, connected by directed edges representing feasible moves between nodes, so that the graph with these elements, the "Sampled PPS Graph", will support reaching.

This graph would ideally be created while motor babbling, making random changes to the joint angle vector, and sensing infeasible points by detecting collisions with self or the environment while preventing damage to the robot. Unfortunately, our physical robot (Fig. 1) lacks touch sensors and could damage itself. So, during graph creation only, we use the MoveIt! Motion Planner [16] with geometric models of the robot and the environment, to test randomly generated points in the robot's configuration space for feasibility. This safety check is no longer needed or used after this process of graph creation. Once the PPS Graph is completed, the agent operates under the assumption that all moves along its edges are feasible, and does not have access to any motion planner or geometric models for Experiments 2-6.

Each new node n_i is defined by a joint angle vector $q_i = Q(n_i)$, randomly generated and checked for feasibility. Once the robot reaches that state, it collects the visual sensory vector $P(n_i)$. No foreground objects are present while the graph is built so that each stored $P(n_i)$ contains only the robot and static neutral background. This facilitates the identification of foreground objects in later visual percepts.

Since nodes are feasible by construction, we assume that relatively short edges represent feasible motions using linear interpolation. Therefore, to create the set of edges, each node is connected to its five nearest neighbors in joint configuration space. Additional edges are added as necessary to ensure the graph is symmetric. Each edge is annotated with $G(e_{ij})$, which is found by the convex hull approximation due to the large number of videos that would otherwise be required. The resulting Sampled PPS Graph is visualized spatially in Fig. 2. Creating it requires approximately 10

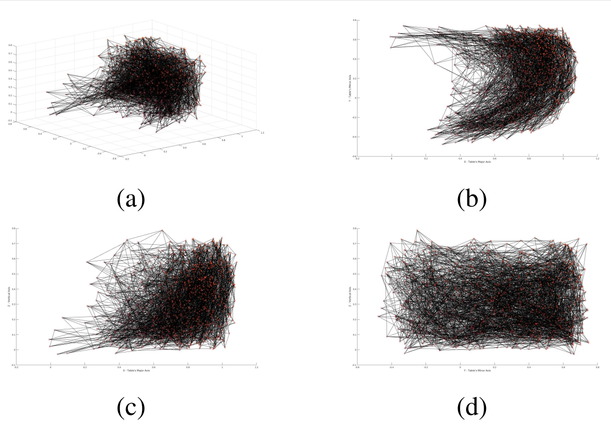


Fig. 2. (a) The Sampled PPS Graph with 1001 nodes and 6460 directed edges, with nodes plotted according to the 3D positon of the end-effector. *The agent only has access to a topological abstraction of this structure.* (b)-(d) 2D projections of the Sampled PPS Graph. Note that the random configuration space sampling procedure has produced a dense, well-covering structure, especially in the region most natural to sweep the arm through.

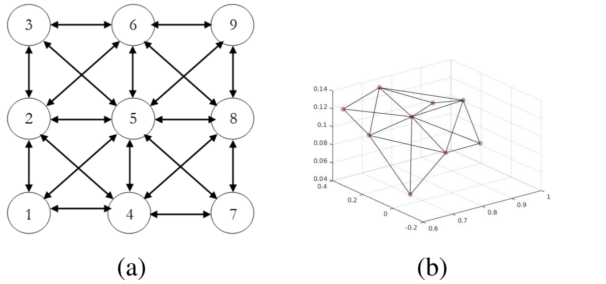


Fig. 3. The Learning Graph: (a) The topology of the abstract model used by the agent. (b) The metrical structure of the graph, illustrated by plotting the 3D end-effector positions at each node. This structure is not available to the agent.

hours on the physical robot (Fig. 1). The graph appears qualitatively to have the density and coverage to support reaching in PPS. Experiment 6 will assess quantitatively whether the graph supports reaching.

B. Creating the Learning Graph

Our model relies on the observation of unusual events to motivate learning, but in the large state space of the Sampled PPS Graph, such events could be prohibitively rare. (Note that human infants learn to reach over several months, and thousands of arm motions.) To demonstrate the learning process with an accelerated pace, we provided the agent with a smaller Learning Graph for use in Experiments 2-5.

The Learning Graph has 9 nodes generated by manually guiding the robot arm to safe poses and recording each q_i and p_i. These poses place the gripper in a near-planar region parallel to and slightly above a table on which objects will be placed, which was also done to increase the occurrence of unusual events. A set of 40 feasible edges were defined manually, and each $G(e_{ij})$ is determined from videos. The Learning Graph is shown in Fig. 3.

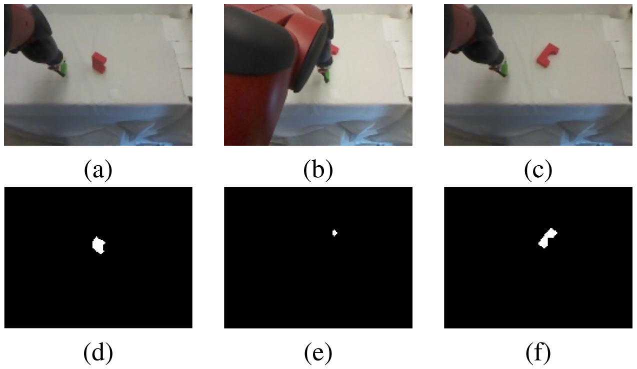


Fig. 4. (a)-(c): The visual observations before each phase of forward-back-forward traversal of $e_{4,7}$ made to facilitate detection of persistent events, or "Bumps". Intuitively, the block has been knocked over since its appearance is significantly different in (a) and (c), despite the same gripper appearance. (d)-(f): The sequence $O_k(n_4, 3) \cdots O'_k(n_7, 3) \cdots O''_k(n_4, 3)$. Using the threshold learned in Experiment 2, the agent can identify that a Bump occurred since $IOU(O_k(n_4, 3), O''_k(n_4, 3)) < a_3$, so A_3, a sufficient condition for a Bump, is true.

The results of Experiments 2-5 demonstrate that the agent learns to predict and perform reliable reaches in the Learning Graph. It is also important that knowledge gained from experience in the Learning Graph generalizes to allow reaching with similar skill in more representative PPS Graphs. We quantitatively support this generalization with a comparison of results from Experiments 5 and 6.

C. Identifying Foreground Objects

When the agent moves along graph edges in the open workspace, the world is static (except for the robot arm), and for each new visual observation $P'(n_i)$ made at n_i, $P'(n_i) \approx P(n_i)$, the stored observation. However, when a distinctive object is added to the workspace, $P'(n_i) \not\approx P(n_i)$, with changes insufficiently explained by noise. The new appearance of the workspace is explained by segmenting connected components of significantly changed pixels in corresponding camera images as a set of $K \geq 1$ foreground objects. These objects may be either *targets*, which are desirable to Bump, or *obstacles*, which are not. When $K > 1$, the objects are placed in the environment and observed separately before being observed in the environment concurrently.

The robot must efficiently distinguish moves that displace an object and moves that merely change the object's appearance by occlusion, using only its visual percepts. Therefore, a move along an edge e_{ij} is performed with a forward-back-forward pattern, starting at n_i and traveling to n_j, then back to n_i, and forward to n_j again. Let the notation $O_k(n_i, c)$ denote the binary image of object k, as seen from camera c when the robot is in state n_i. The forward-back portion of the motion pattern for e_{ij} would then yield the sequence

$$O_k(n_i, c) \cdots O'_k(n_j, c) \cdots O''_k(n_i, c) \quad (3)$$

for each object k and camera c. An example of this sequence of observations and the resulting sequence of binary images is given in Fig. 4.

Intuitively, we can make the required discriminations using the IOU feature (Equation 2):

$$IOU(O_k(n_i,c), O'_k(n_j,c)) \approx 1 \quad \Rightarrow \quad \text{No change} \tag{4}$$
$$IOU(O_k(n_i,c), O'_k(n_j,c)) \ll 1 \quad \Rightarrow \quad \text{Change} \tag{5}$$

This comparison determines the type of a sensed change:

$$IOU(O_k(n_i,c), O''_k(n_i,c)) \approx 1 \quad \Rightarrow \quad \text{Transient} \tag{6}$$
$$IOU(O_k(n_i,c), O''_k(n_i,c)) \ll 1 \quad \Rightarrow \quad \text{Persistent} \tag{7}$$

Experiments 2 and 3 make these distinctions operational, and determine what evidence best indicates the occurrence of a persistent event. Experiment 5 demonstrates how to use perceptual features to predict which motions will reliably cause a persistent event.

D. Experiment 2 - Identifying Unusual Events with a Foreground Object during Random Motion in the Learning Graph

With a single object in the environment (i.e. $K = 1$), the agent follows 72 random edges (216 moves) in the Learning Graph, recording each $O_k(n,c)$. In this work, the agent focuses on the relationship between $O_k(n_i,c)$ and $O''_k(n_i,c)$, where differences indicate a persistent change. Differences between $O_k(n_i,c)$ and $O'_k(n_j,c)$ that do not exist between $O_k(n_i,c)$ and $O''_k(n_i,c)$ are transient changes, often occlusions or shadows from different gripper placement. Since $O'_k(n_j,c)$ is not processed, changes that occur during the "back" portion of the movement along e_{ji} and during the "forward" motion are indistinguishable to the agent. Event detection is limited in granularity to each edge (three moves), and occurs after the agent observes $O''_k(n_i,c)$.

The agent measures the similarity between object silhouettes before and after a move by $IOU(O_k(n_i,c), O''_k(n_i,c))$, observing the set of values shown in the histograms in Fig. 5. The distribution of IOU values for each camera is bimodal with a clear separation between the rare and common clusters of data points. $a_c \in [0,1]$ for each $c \in \{1,2,3\}$ is selected as the threshold value of $IOU(O_k(n_i,c), O_k(n_i,c))$ with the same z-score for the normal distribution fit to the rare cluster with $IOU(O_k(n_i,c), O''_k(n_i,c)) \ll 1$ and the normal distribution fit to the common cluster with $IOU(O_k(n_i,c), O''_k(n_i,c)) \approx 1$. For each $c \in \{1,2,3\}$,

$$A_c \equiv [IOU(O_k(n_i,c), O''_k(n_i,c)) < a_c]\,. \tag{8}$$

A_c is false in the special case $O_k(n_i,c) = O''_k(n_i,c) = \emptyset$, where the IOU is undefined. The agent makes a second pass through the $IOU(O_k(n_i,c), O''_k(n_i,c))$ data, identifying a small number of persistent, unusual events, which we will call "Bumps", by the criterion $A_1 \vee A_2 \vee A_3$, and a larger number of common case observations, consisting of nonevents and transient events that do not meet this criterion.

E. Experiment 3 - Learning the Reliability of Bump Prediction with a Set of Positive Intersection Features

Observing a small set of Bumps in the previous experiment motivates the agent to predict which move or moves in a given situation are most likely to cause a Bump. For each e_{ij} that was traversed, and object k, the agent generates Boolean intersection features

$$B_c \equiv [G(e_{ij},c) \cap O_k(n(k),c) \neq \emptyset], \tag{9}$$

where $n(k)$ is the node where the agent had most recently observed object k at the time of deciding whether to follow e_{ij}. Since the agent followed individual edges randomly in Experiment 2, $n(k) = n_i$. Since $G(e_{ij},c)$ was observed during the initial formation of the graph, and $O_k(n(k),c)$ is observed before a move is decided or performed, B_c can be determined prior to a move, and B, the set of positive B_c features, can be used to predict a Bump in advance. Fig. 6 shows, for each possible B, the portion of resulting observations that were classified as Bumps, yielding initial estimates of the conditional probability, $P_0(Bump \mid B)$.

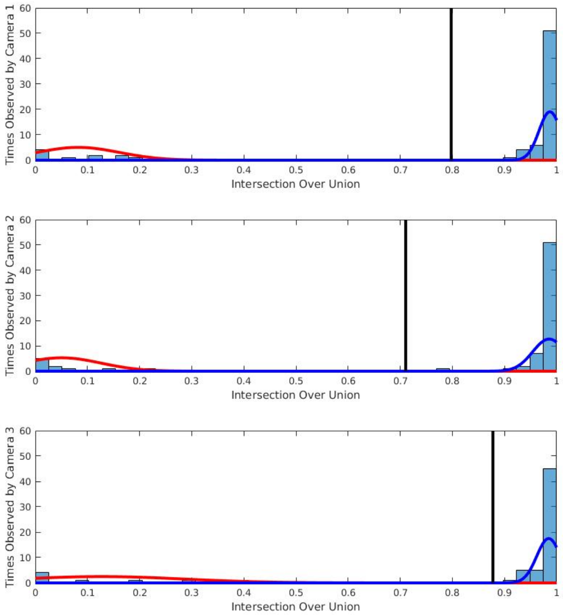


Fig. 5. Learning thresholds for event detection: The agent determines $IOU(O_k(n_i,c), O''_k(n_i,c))$ for each camera c for moves along 72 random edges. Creating histograms of these values shows well-separated high and low clusters for the dataset of each c. Each plot marks the significant change threshold a_c, the IOU value with the same absolute value z-score for each cluster's normal distribution. These thresholds allow agent classification of observed moves by the $Bump \equiv A_1 \vee A_2 \vee A_3$ criterion, where $A_c \equiv [IOU(O_k(n_i,c), O''_k(n_i,c)) < a_c]$.

The agent's current experience suggests that traversing any edge with $B = \{B_1, B_2, B_3\}$ will always cause a Bump. Further, $P_0(Bump \mid B)$ is much lower for any other B, so $\{B_1, B_2, B_3\}$ is by far the most reliable predictor of a Bump at this time. The agent continues to update these conditional probability approximations $P(Bump \mid B)$ from experience gathered during Experiment 4 and additional practice, ending with those recorded in the $P_1(Bump \mid B)$ column of Fig. 6.

F. Experiment 4 - Shortest Reaches in the Learning Graph

A reach is a traversal of one or more consecutive graph edges that intends to cause a Bump with a target object on

B	$P_0(Bump \mid B)$			$P_1(Bump \mid B)$		
$\emptyset$	0	=	0/2	0.214	=	6/28
$\{B_1\}$	0.067	=	1/15	0.171	=	7/41
$\{B_2\}$	0	=	0/19	0.096	=	9/94
$\{B_3\}$	0	=	0/0	0	=	0/10
$\{B_1, B_2\}$	0.071	=	2/28	0.342	=	39/114
$\{B_1, B_3\}$	0	=	0/0	0.458	=	11/24
$\{B_2, B_3\}$	0	=	0/1	0.727	=	8/11
$\{B_1, B_2, B_3\}$	1.0	=	7/7	0.840	=	137/163
Any	0.139	=	10/72	0.447	=	217/485

Fig. 6. The agent's approximation of the conditional probability of a persistent event, a "Bump", given a set B of positive intersection features after 72 random moves in Experiment 2 (P_0) and after an additional 413 moves during goal-oriented practice (P_1). In both cases, the full set of features $B = \{B_1, B_2, B_3\}$ gives the highest probability of a Bump.

the final move. It can fail if no Bump occurs, or have partial success if the Bump occurs "accidentally" before the final move. Full success requires that the Bump occurs on the final move of the trajectory, with no previous accidental Bumps.

The agent is sequentially presented with 100 single-object environments ($K = 1$), using one of four objects with varying size and shape (shown in Fig. 1) in 25 trials each. The object is placed at random (x, y) coordinates in the 43cm x 30cm grid enclosed by the projection of the Learning Graph onto the table's surface, and a random orientation θ between 0 and 2π. The robot begins each trial at n_1, which has no initial occlusions of the object. The agent plans the full trajectory to reach to this target before leaving n_1, so $n(k) = n_1$ in the determination of each B_c. The set of candidate final edges E^* with feature set $\arg\max_B P(Bump \mid B)$ is generated, with the procedure visualized in Fig. 7. If $E^* = \emptyset$, sets of edges with B yielding progressively lower $P(Bump \mid B)$ are included until $E^* \neq \emptyset$. The agent gives each edge in the Learning Graph a uniform cost, and searches the graph using Dijkstra's algorithm to find the lowest cost path ending with an edge in E^*. The agent traverses each edge of this trajectory with the forward-back-forward pattern, checking for Bumps and updating $P(Bump \mid B)$.

The success rates of these reaches are presented in Fig. 8. In a majority (65%) of trials, deliberate Bumps are caused by executing these trajectories. Attempting reaches allows the agent to gain more experience, and hence to gain more information about Bump prerequisites. In particular, the updates to $P(Bump \mid \{B_1, B_2, B_3\})$ show that $\{B_1, B_2, B_3\}$ is still the most reliable predictor, but that $\{B_1, B_2, B_3\}$ is not sufficient to guarantee a Bump (see Fig. 6). Reaches with $B = \{B_1, B_2, B_3\}$ fail especially often when the object is large, and with very small intersection regions near the object boundary. The failure cases of Experiment 4 motivate the generation of more features to predict more accurately the probability of a Bump using a specific edge in E^*.

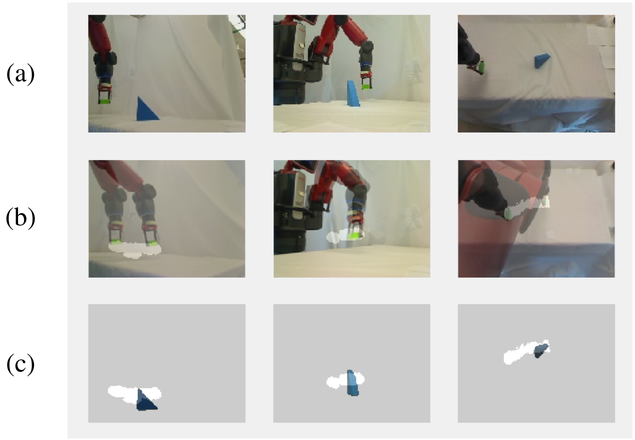


Fig. 7. The process for determining whether an edge e_{ij} is a member of E^*, the set of candidate final edges for a reach trajectory. (a) The agent records $P'(n_1)$, the current visual state of the environment with the object present. The object silhouette in each $P'(n_1, c)$ is identified as the largest connected component of pixels with appearances significantly changed from their appearance in $P(n_1, c)$. (b) The agent recalls each $G(e_{ij}, c)$, the region of the video frames from camera c the gripper passes through while moving from n_i to n_j. $G(e_{ij})$ is visualized here in white, superimposed with $P(n_i)$ and $P(n_j)$. (c) The truth of each intersection feature B_c is determined by whether the object silhouette (shown here in color) and $G(e_{ij}, c)$ intersect for each c. All are true, so $B = \{B_1, B_2, B_3\}$, and e_{ij} is included in E^* because this maximizes $P(Bump \mid B)$.

G. Generation of Additional Features for Trajectory Planning with Naïve Bayes

Experiment 4 shows that $P(Bump \mid \{B_1, B_2, B_3\}) \neq 1$, and that no set of features B can guarantee a Bump. It becomes desirable for the agent to differentiate edges in E^* that lead to deliberate Bumps from those that will not. The agent has records of $O_k(n(k), c)$ and the final edge $G(e_{ij})$ for each of the 100 attempted reaches, of which 67 were perceived by the agent as fully successful and 33 were perceived as partial success or failure.

To attempt to improve its prediction of successful reaches, the agent creates 21 new feature values that capture additional properties of the intersection region, such as its size and centrality in the silhouettes. For each edge e_{ij} and object k, the agent generates the following features D_{fc}, which are computed from $O = O_k(n(k), c)$ and $G = G(e_{ij}, c)$:

$$\begin{aligned}
D_{1,c} &= |O \cap G| \\
D_{2,c} &= |O \cap G| / |O| \\
D_{3,c} &= |O \cap G| / |G| \\
D_{4,c} &= IOU(O, G) = |O \cap G| / |O \cup G| \\
D_{5,c} &= \arg\min_s [erode(O, square(s)) \cap G = \emptyset] \\
D_{6,c} &= \arg\min_s [erode(G, square(s)) \cap O = \emptyset] \\
D_{7,c} &= \arg\min_s [erode(O, square(s)) \\
&\qquad\qquad \cap\, erode(G, square(s)) = \emptyset]
\end{aligned} \tag{10}$$

Given two binary images O and G, these are natural relational features that can be generated automatically without prior knowledge of the domain. These features can serve as predictors since they are constructed from information gathered without first having to perform a motion.

Classifier	Full Success	Partial Success	Either Success
Agent Perception	67%	16%	83%
Ground Truth	65%	8%	73%

Fig. 8. The success rate of reaches in Experiment 4, planned using simple intersection features in environments with one target over 100 trials, as determined by agent perception and an experimenter ground truth observation that corrects for false positives and negatives. Full success differs from partial success in that the target object must be Bumped on the final edge of the reach trajectory only.

The agent fits normal distributions to the values of each D_{fc} for the 67 successful trials and for the 33 unsuccessful trials. Most of these distributions are similar between the two cases, and no single feature can reliably distinguish between the success and failure cases. However, treating all 21 features as weak sources of information, the agent can use a Naïve Bayes classifier to predict whether a given edge and object combination will result in a Bump, achieving 81% accuracy over the initial set of 100 trials.

For a given edge e_{ij} and object k, we let D (or, when necessary, $D_{e_{ij},k}$) denote the set of 21 D_{fc} features, $Bump$ denotes the event of motion along e_{ij} bumping object k, and the prior probability $P(Bump)$ is the value from Fig. 6. The Naïve Bayes classifier is then:

$$P(Bump \mid D) = \alpha P(Bump) \prod_{f,c} P(D_{fc} \mid Bump). \quad (11)$$

Let the edges of a trajectory T with $length(T) = L$ be denoted by e_l for $l \in \{1, \ldots, L\}$. A fully successful reach requires a deliberate Bump, done with e_L, and no Bumps with any of e_1 through e_{L-1}. Therefore, for a fully successful reach to a target object $k = 1$,

$$P(Full \mid T) = P(Bump \mid D_{e_L,1}) \prod_{l=1}^{L-1} (1 - P(Bump \mid D_{e_l,1})). \quad (12)$$

Avoiding obstacles requires that no e_l causes a Bump, so for environments with obstacle objects $k \in \{2, \ldots, K\}$,

$$P(Avoid \mid T) = \prod_{k=2}^{K} \prod_{l=1}^{L} (1 - P(Bump \mid D_{e_l,k})). \quad (13)$$

This formulation allows the same knowledge to support reaching and obstacle avoidance, and is successfully used in Experiments 5 and 6, where a single trajectory T is chosen to maximize the probability of accomplishing both goals,

$$P(Both\ Goals \mid T) = P(Full \mid T) \cdot P(Avoid \mid T). \quad (14)$$

H. Experiment 5 - Reaches in the Learning Graph while Avoiding Obstacles and Accidental Target Bumps

The agent is given 25 randomized $K = 2$ environments, and performs two trials in each, one for each assignment of target and obstacle roles. The agent uses the Learning Graph, and the set E^* is found as before. The trajectory

$$T^* = \arg\max_T P(Both\ Goals \mid T) \quad (15)$$

is found from the set of all trajectories T that begin at starting pose n_1 and end with an edge in E^*. Given D, the trajectory T^* has the highest probability of a fully successful reach that also avoids collisions with obstacles.

Task	Success Rates by Graph Used	
	Learning Graph	Sampled PPS Graph
Reach Target	90%	90%
Avoid Obstacle	86%	74%
Both Goals	76%	68%

Fig. 9. Performance of the Agent over 50 trials at executing a single trajectory to reach to a target and avoid an obstacle, according to the experimenter judged ground truth criterion. It is clear that the introduction of an obstacle does not prevent the agent from making effective reaches in either graph. The high level of success in the Sampled PPS Graph demonstrates that the learning from the Learning Graph transfers well, and applies to a general graph.

The results in the Learning Graph column of Fig. 9 show that the rate of fully successful reaches have increased to 90% from 65% in Experiment 4, even with the added challenge of obstacle avoidance, demonstrating the benefits of the intersection region features D_{fc} for planning reaches. We believe that this large performance increase is due to Naïve Bayes over the D_{fc} features being able to reduce occurrence of accidental Bumps and the selection of final edges with minimal intersections.

The agent's ability to avoid the obstacle in 86% of trials, and to succeed at both reaching and avoidance with the same trajectory 76% of the time, justifies the Naïve Bayes formulation for selecting T^*. Achieving both goals while planning with the same set of features, and without explicitly training to avoid obstacles, implies that learning transfers well from reaching to avoidance tasks. We continue to analyze the strength of this formulation and the generality of these features in the next experiment.

I. Experiment 6 - Generalized Reaching and Obstacle Avoidance in the Sampled PPS Graph

We now return to the questions of the density and coverage of the Sampled PPS Graph. We also quantitatively test whether the agent's Learning Graph experience supports reaching in the randomly-generated Sampled PPS Graph, which would be sufficient to show that the learned features are general to any graph representation of peri-personal space. This result will also demonstrate that planning in a larger, more dense graph is tractable and yields similar success rates. We conduct an experiment using the Sampled PPS Graph generated in Experiment 1, and control the rest of the experimental factors, notably using the same 25 pairs of trials (with identical object placements) as Experiment 5 and the same procedure for generating T^*. The results are presented in the Sampled PPS Graph column of Fig. 9.

The agent's 90% rate of fully successful reaches demonstrates that the tabletop grid is well-covered. Further, the 74% obstacle avoidance rate suggests that there is also enough density in the graph that trajectories were not often forced to include edges that Bump the obstacle in order to Bump the target. Since the locations of the Sampled PPS Graph nodes and edges are not correlated to locations in the grid, it follows that the graph has sufficient coverage and density in arbitrary subregions of PPS, and thus the entirety of PPS.

The number of successful reaches is unchanged from Experiment 5, and the drop in obstacle avoidance success rate when using the Sampled PPS Graph may be explained by limitations of the convex hull approximations for each $G(e_{ij}, c)$ used with the Sampled PPS Graph, which do not capture the extremes of the region caused by curved or under-damped motion. We conclude that the simple Experiment 1 procedure has produced a representation of PPS that supports the planning of safe and reliable reaches using the B and D features generated from Learning Graph experience.

III. DISCUSSION AND CONCLUSIONS

An important foundational question about intelligence is, "*Where does knowledge of space come from?*" The default answer embodied by most work in AI and robotics is, "*It is built in by an intelligent designer.*"

The Spatial Semantic Hierarchy [10], [2] describes how knowledge of large-scale space can be learned from sensorimotor experience, given pre-existing knowledge of local control laws and local SLAM methods for small-scale space. We have also shown how a robot learning agent without prior knowledge of its sensors and effectors, can use domain-independent statistical methods to learn (a) the structure of its sensor arrays; (b) sensory features and control laws; (c) the representational basis for local SLAM algorithms; and (d) objects distinguished from the static environment and actions that can be applied to them [13], [11].

In this paper, we present initial steps toward a computational model of how a learning agent can learn another important kind of space: the *peri-personal space* within which actions like *reaching* and *grasping* take place. We describe a joint visual-proprioceptive representation for peri-personal space — the *PPS Graph* — that can be created without prior knowledge (or subsequent reconstruction) of its 3D structure, or of the 3D interpretation of visual input, or the kinematic implications of sensed joint angles for manipulator configuration.

Experiments with a small model *Learning Graph* demonstrate that the simple PPS Graph representation can support discovery of unusual events (i.e., reaching) in which the robot's manipulator interacts with nearby objects in the environment. They evaluate search methods for finding more reliable trajectories for reaching nearby objects. This level of competence corresponds roughly to the *gross reaching* level observed in human infants. An experiment with the *Sampled PPS Graph* demonstrates that a dense graph can provide good coverage of the local peri-personal space and that trajectory-search methods scale up reliably from the small Learning Graph to the much larger Sampled PPS Graph.

Future work will extend this to the more dextrous *fine reaching* level, which allows *grasping* to be identified as a rare action, thanks to the *palmar reflex* [8]. Reaching, grasping, and then moving objects from place to place are the basis for mastery of peri-personal space.

ACKNOWLEDGMENT

We gratefully acknowledge valuable advice from Chad Jenkins. This work has taken place in the Intelligent Robotics Lab in the Computer Science and Engineering Division of the University of Michigan. Research of the Intelligent Robotics lab is supported in part by grants from the National Science Foundation (IIS-1111494 and IIS-1421168).

REFERENCES

[1] G. Baldassarre and M. Mirolli, editors. *Intrinsically Motivated Learning in Natural and Artificial Systems*. Springer, 2013.

[2] P. Beeson, J. Modayil, and B. Kuipers. Factoring the mapping problem: Mobile robot map-building in the Hybrid Spatial Semantic Hierarchy. *International Journal of Robotics Research*, 29(4):428–459, 2010.

[3] N. E. Berthier. The syntax of human infant reaching. In *Unifying Themes in Complex Systems: Proceedings of the Eighth Int. Conf. on Complex Systems*, volume Volume VIII of *New England Complex Systems Institute Series on Complexity*, pages 1477–1487. NECSI Knowledge Press, 2011.

[4] N. E. Berthier, M. T. Rosenstein, and A. G. Barto. Approximate optimal control as a model for motor learning. *Psychological Review*, 112(2):329–346, 2005.

[5] E. Chinellato, M. Antonelli, B. J. Grzyb, and A. P. del Pobil. Implicit sensorimotor mapping of the peripersonal space by gazing and reaching. *IEEE Trans. on Autonomous Mental Development*, 3(1):43–53, 2011.

[6] R. K. Clifton, D. W. Muir, D. H. Ashmead, and M. G. Clarkson. Is visually guided reaching in early infancy a myth? *Child Development*, 64(4):1099–1110, 1993.

[7] D. Corbetta, S. L. Thurman, R. F. Wiener, Y. Guan, and J. L. Williams. Mapping the feel of the arm with the sight of the object: on the embodied origins of infant reaching. *Frontiers in Psychology*, 5(576), June 2014. http://dx.doi.org/10.3389/fpsyg.2014.00576.

[8] Y. Futagi, Y. Toribe, and Y. Suzuki. The grasp reflex and Moro reflex in infants: Hierarchy of primitive reflex responses. *International Journal of Pediatrics*, 2012(191562), 2012. doi:10.1155/2012/191562.

[9] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[10] B. Kuipers. The Spatial Semantic Hierarchy. *Artificial Intelligence*, 119:191–233, 2000.

[11] J. Modayil and B. Kuipers. The initial development of object knowledge by a learning robot. *Robotics and Autonomous Systems*, 56:879–890, 2008.

[12] Jean Piaget. *The Origins of Intelligence in Children*. Norton, New York, 1952.

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

[14] P. Savastano and S. Nolfi. A robotic model of reaching and grasping development. *IEEE Trans. on Autonomous Mental Development*, 5(4):326–336, 2013.

[15] M. W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. Wiley, 2006.

[16] I. A. Sucan and S. Chitta. Moveit! [Online] Available http://moveit.ros.org. Accessed 3-25-2016.

[17] E. Thelen, D. Corbetta, K. Kamm, J. P. Spencer, K. Schneider, and R. F. Zernicke. The transition to reaching: mapping intention and intrinsic dynamics. *Child Development*, 64(4):1058–1098, 1993.

[18] E. Ugur, Y. Nagai, E. Sahin, and E. Oztop. Staged development of robot skills: behavior formation, affordance learning and imitation with motionese. *IEEE Trans. on Autonomous Mental Development*, 7(2):119–139, 2015.

[19] C. von Hofsten and L. Rönnqvist. The structuring of neonatal arm movements. *Child Development*, 64(4):1046–1057, 1993.