In International Conference on Development and Learning (ICDL),
Bloomington, IN, May 2006.

Behavior-Based Segmentation of Demonstrated Tasks

Nathan Koenig
Interaction Laboratory
University of Southern California
University Park, Los Angeles, California, USA

Maja J Matarić
Interaction Laboratory
University of Southern California
University Park, Los Angeles, California, USA

***Abstract*— Robot learning from demonstration presents several challenges. Given a task demonstration, the robot must sense, understand, and learn appropriate task attributes. We propose a method for automatic segmentation of a complex demonstration into an ordered set of simpler behaviors. These behaviors present a significantly less complex domain for learning. Our method is based on empirically derived attributes of tasks and simple mathematical transformations that results in a fast and intuitive mechanism for automatic task segmentation. The method is validated on a simulated Pioneer 2 DX platform across four demonstrations that vary in environmental complexity, task complexity, and task strategy.**

***Index Terms*— Teaching, demonstration, behavior segmentation**

I. Introduction

Teaching by demonstration is an elementary instructional tool that people use instinctively. When describing a task we tend to actively go through the motions with the understanding that the person watching will learn from the demonstration. The intuitive nature of such teaching makes it a compelling research topic toward robot programming by demonstration. Current robot programming techniques generally involve hand coding routines and require significant experience, education, and time. However, a robot capable of understanding a task demonstration significantly reduces the requirements placed on the human programmer. By removing the complexity from the human and placing the burden on the robot to act and understand within an environment creates a more versatile and friendly tool.

As a first step toward addressing the problem of understanding demonstrations, we propose a segmentation algorithm that partitions a demonstration into significant components. Our method is based on the observation that most tasks can be decomposed into a series of simpler *behaviors*. These behaviors span smaller segments of time than the complete demonstration, and focus on specific aspects of the task. By partitioning the demonstration into easier-to-understand segments, the process of learning becomes simplified.

A segmentation algorithm to fit our needs must have a few specific attributes. It must be fast and capable of running online, rather than in a batch process. This allows the robot to perform segmentation during teaching, and removes the need for waiting after a demonstration to process the information. The algorithm must also operate on data from arbitrary features. By this we mean the algorithm cannot be limited to process only specific sensory data.

In this paper we do not describe a method for learning the behaviors themselves. Instead we focus on the problem of identifying where behaviors are in a demonstration sequence. However, one can think of using regression techniques, reinforcement learning, or neural networks as means of learning control rules for the behaviors. We thus leave learning of the internal controller for a behavior to future work and focus on finding temporal boundaries for behaviors in a task demonstration.

II. Related Work

Our work offers a new starting point for programming by demonstration, which is a well-studied topic for user interfaces, graphics applications, and office automation. Each of these domains involves learning action sequences by observing user actions. These action sequences are then applied when similar situations are encountered. In robotics, programming by demonstration has been applied to a variety of tasks. In industrial robotics [13], focus was placed on parameter learning of elementary operations. Besides parameter learning, robots have also learned behavior network representations for complex tasks using an interactive teaching methodology [15].

Imitation learning in humanoid robots is concerned with learning parametric models and policies of motion from human demonstrations. An articulated arm successfully learned an appropriate policy to balance a pole based on a reward function and task model [2]. Biped humanoids have learned to achieve human-like locomotion through motion primitives of demonstrated trajectories [14]. General formalisms for performance metrics on humanoid imitation tasks have also been studied [4]. This form of imitation allows articulated robots to learn complex gestures and motions by observing a human.

Our work relies on segmenting sequences of data into distinct behaviors. The first approach we studied used an entropy based measure, the Jensen-Shannon divergence [10]. This measure was successfully used to segment traces of human movement in the plane, gathered by a laser range finder, into anomalous interactions [16]. However, we found the that method does not lend itself well to data sequences with a large range, and it cannot be used in an on-line manner. Instead we turned to a simple derivative based approach, where analysis of a data stream's gradient marked behavior boundaries at local minima.

Work similar to sequence segmentation involves signal approximation through the use of line segments. This technique has been studied in many fields, and is known also as piecewise linear approximation [3], [7], [8]. The approximation attempts to reduce the complexity and size of large data sets. Financial applications and computer graphics use this technique for pattern matching, contour tracing and boundary analysis, and data compression [5], [6]. Piecewise linear approximation is generally run offline and can be fairly complex. While our work is also focused boundary analysis, we require an algorithm suitable for fast online analysis of numerous data sequences.

III. Overview of Approach

The goal of this work is the identification of boundaries, *segmentation points* or *transition points*, between behaviors in a task demonstration. If these transition points are identified properly, the problem of learning complete tasks transforms into a problem of learning simpler behaviors. This has two key benefits. The first is the reduction of complexity for the learner, the second the potential for reuse of simple behaviors toward learning new tasks.

We assume that the robot has no prior knowledge about a demonstrated task, and therefore must determine where behaviors lie from the information in the demonstration. The robot does have access to its entire *feature space*, which is composed of sensory and model-based information. The later consists of processed data. The potential size of this feature space is large, and is often comprised of information that may be irrelevant to the task at hand. We therefore require the teacher to specify a subset of *salient features* that are most relevant to the task.

This subset generally consists of model-based data as they tend to be less noisy, sparser, and more intuitive for a human user to understand. As an example, if one wishes to teach a robot how to follow a wall then the relevant feature would be the distance to the wall. This feature relies on the robot knowing what a wall is and how to measure distance.

With an appropriate set of features identified, the teacher then guides the robot through the desired task while logs are created containing data from the salient features. The resulting data sequences form the basis upon which behavior segmentation is performed. For each data sequence, a cost function is minimized by splitting the sequence at points of maximal cost. These split points define potential locations for behavior segmentation. Each set of points (one per data sequence) is temporally grouped. Groups of sufficient size define the final segmentation points of the demonstration.

IV. State Representation

As noted, the state of the robot consists of sensory and model-based information. Sensory information consists of inputs from physical sensors on the robot while model-based information consists of processed sensor data and provided data, such as a map of the environment, portion of the map explored, noted important locations within the map, etc. The amount and type of model-based information available to the robot must be determined prior to training.

A subset of the resulting large state data is used by the robot for segmentation. That subset is composed of the most important, i.e., salient features selected by the teacher, prior to training. A *feature* refers to an individual element of the complete state description, such as the robot's position or the current laser range scan. Each feature must have a corresponding analytical representation that makes it amenable to analysis. The time-series of data from each feature over the course of a demonstration form a function.

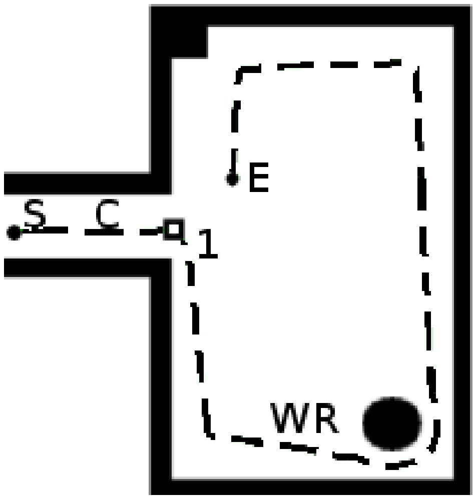
The work in this paper is applied to (but not limited to) the domain of mobile navigation and manipulation. In that context, we make a few assumptions concerning what state information is available to the robot. Specifically, a map of the environment is provided, the robot is capable of localizing itself within the environment, and the robot can detect manipulable objects.

V. Behaviors

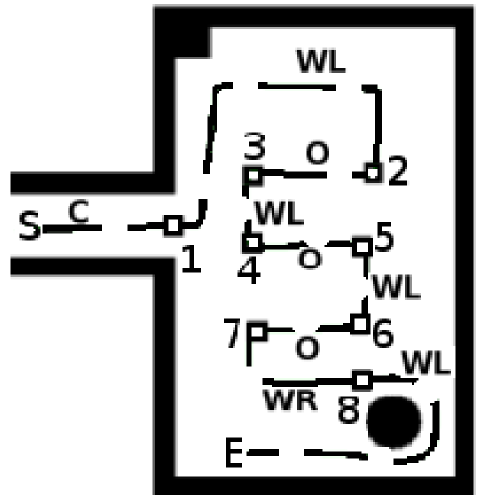
We take a *behavior-based* approach throughout our work [1], [11]. Behaviors are time-extended actions that achieve or maintain a set of goals. These behaviors can describe fairly basic concepts such as `avoid-obstacles` and `follow-wall`, as well as more complex such as `find-box` and `locate-target`. We assume that observed behavior can be segmented into such observable behaviors. In this work, the focus is placed on automatically finding the time extents of such behaviors through analysis of sequential feature data. We do not attempt to label the behaviors themselves. Rather, the segmentation provides a first step toward learning new behaviors and using them to accomplish specific demonstrated tasks.

VI. Human-Robot Interface

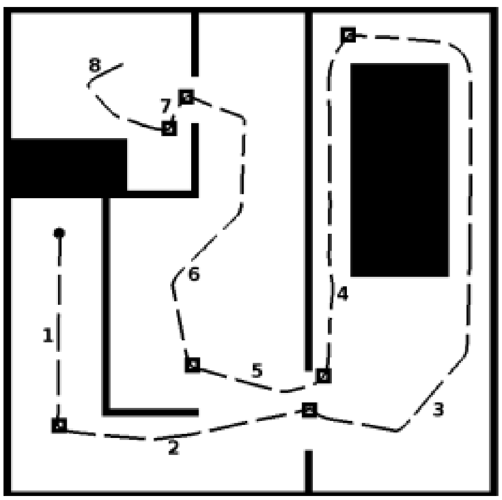
Our method of teaching a robot requires an intuitive interface between the teacher and the student. The interface must inform the teacher of the robot's current state, allow the teacher to instruct the robot, and be simple to use. To meet these demands we rely on a standard personal computer running a graphical user interface (GUI) that processes raw state information and displays it in a meaningful way. The GUI must be able to render all the state information either in image or text format. The rendering method is dependent on the feature type, and varies from an image or animation to a map or a bar meter.



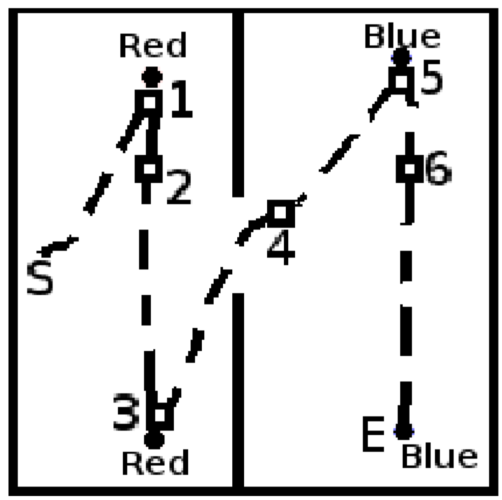
(a) Clockwise exploration



(b) Zig-zag exploration



(c) Multi-room exploration



(d) Pick and place task

Fig. 1. Environment exploration demonstrations. In each figure walls are in black, the path the robot followed is dashed, and the segmentation points are numbered black hollow squares along the path. Segments for the three exploration tasks are labeled with abbreviations for logical behaviors. C = corridor-follow, WR = wallfollow-right, WL = wallfollow-left, O=openspace-traverse. Note that these labels are for improving the readability of the figures; they do not indicate actual behaviors. The start and and of a demonstration are labeled with an S and E.

A mechanism must also exist for controlling the robot during teaching, that maps user input to the robot's actuators in an intuitive way. Since we have assumed that the robot has a map of the environment, which is displayed by the GUI, the user can easily control the robot's position by indicating map locations. These locations are interpolated into a trajectory for the robot to follow. This abstraction allows the teacher to ignore the kinematics and dynamics of the robot hardware, and instead focus on the learning task. Similarly, grippers can be controlled by a set of buttons that indicate opening, closing, and lifting. Such interfaces are needed for each of the robot's actuators.

VII. Segmentation Algorithm

Our algorithm for the determination of behavior boundaries relies on having one or more sequences of concurrent feature data as input. This information is easily collected from data logs generated during a training exercise in which a teacher commands a robot to perform a specific task. We assume the robot has the capability to store feature data, and that the teacher is capable of commanding the robot to perform the desired task. Finally, the algorithm we describe is performed as a batch process, but can be run in real time with few changes.

A data sequence s consists of N normalized samples $x(1), x(2), ..., x(N)$. We define a segment of the data sequence s as $s(a, b)$ consisting of consecutive samples $x(a), x(a+1), ..., x(b)$ where $a < b$. S_k denotes a complete segmentation of the sequence s into k consecutive non-empty

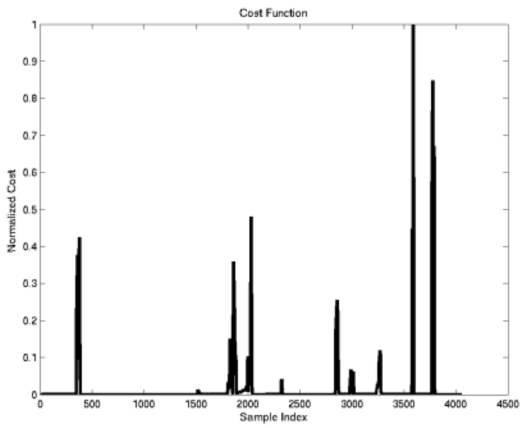


Fig. 2. Variance-based cost function.

Algorithm 1 Segmentation Pseudo-code

Require: $FS = s_i, s_i + 1, ..., s_k$
Require: $C = \{\phi\}$
Require: $B = \{\phi\}$
Require:
  for all s_i in FS **do**
    - Apply cost function F_{cost} to s_i using a sliding window $w(a, b)$ of length N.
      $s_{i_{cost}} = F_{cost}(w(a, a + N))$
    - Threshold $s_{i_{cost}}$ below mean.
    - Find segmentation point C_i in $s_{i_{cost}}$.
      $C_i(t) = \begin{cases} 1, & \text{if } s_i(t) == local_max \\ 0, & \text{otherwise} \end{cases}$
  end for
  for all C_i **do**
    $B(t) = \begin{cases} 1, & \text{if } \exists\ C_j, t_2 : ||C_i(t) - C_j(t_2)|| < 4 \\ 0, & \text{otherwise} \end{cases}$
  end for
  {Where FS is a set of feature sequences, C_i is a set of segmentation points for sequence s_i, and B is the final set of behavior boundary points.}

---

segments. The boundary points between the k segments are defined as $c_1, c_2, ..., c_{k-1}$, where $0 < c_1 < c_2 < ... < c_{k-1}$. Formally, $S_k = s_1, s_2, ..., s_k$ and $s_1 = s(1, c_1), s_2 = s(c_1 + 1, c_2), ..., s_k = s(c_{k-1} + 1, N)$.

A task demonstration begins by placing the robot in the relevant environment with a given set of salient features. The teacher then uses the GUI to send commands to to the robot until it reaches the desired goal state. Commands are high level in nature, such as an absolute position to reach, opening or closing a gripper, etc. During this demonstration process, the robot automatically stores all information from the user-defined time-series of salient features. This results in M concurrent features sequences $FS = s_1, s_2, ..., s_M$, where s_i denotes a sequence.

Segmentation involves manipulation of the stored feature data. Each feature sequence s_i of size N is decomposed into a set of K behavior segments $S_{i_k} = s_i(1, c_1), s_i(c_1 + 1, c_2), ..., s_i(c_{k-1}, N)$ partitioned by $k - 1$ boundary points $C_k = c_{i_1}, c_{i_2}, ..., c_{i_k}$. Each feature sequence is initially processed individually. The final step in the algorithm correlates data from all sequences. This step is skipped when only one feature was monitored.

Initially a cost function F_{cost} is applied to a sliding window $w_i(a, b)$ that is passed over each feature sequence. The cost function we use is the variance in the feature data, and the window size is empirically set to 5 data samples.

$$F_{cost}(w_i(a, b)) = \frac{1}{n}\sum_{j=a}^{b} x_i(j)^2 - \left(\frac{1}{n}\sum_{j=a}^{b} x_i(j)\right)^2$$

The result is a new function that peaks at the maximum local cost, which marks a segmentation point in the sequence, as shown in Figure VI. With the sliding window maximum values occur at points of maximum variance withing the window. We thresholded the function below the mean value in order to eliminate noise, and used the thresholded version to compute the peak/segmentation points. These points mark distinct changes in the given feature, and therefore indicate a high likelihood of a behavior transition.

Each feature sequence is processed as per above, resulting in multiple sets of possible behavior transition points. These points are then correlated across features. Time points that mark behavior transitions have high consistency among the monitored features. We required that at least two separate features agree on a transition point, with some degree of error, for it to be considered valid. We set the allowed error to four seconds, based on empirical evidence. If only one feature had been monitored, no correlation checks were performed. Furthermore, during the course of a demonstration, there may be periods when only one feature provides valid data. For example, consider a robot that is given distances to walls on both right and left sides as features. During periods when it detects only only one wall, on either side, only the data from that feature will be correlated.

Pseudo code for the complete segmentation algorithm is shown in Algorithm 1. The result of the algorithm is a set of locations that mark boundaries between behaviors. With this information, machine learning can be applied to derive controllers for each behavior based on the observed motor commands and changes in feature space between transition points. Furthermore, the segmentation points also specify when the robot should transition from one behavior to another.

VIII. Experimental Design

To validate our algorithm, we performed an experiment involving a mobile robot platform and two tasks. The first task requires teaching the robot how to explore an indoor environment, and the second how to pick up and place colored boxes. These two tasks were chosen to demonstrate the ability of our algorithm to work with both simple and complex tasks that differ significantly.

Our experiments to data were performed in simulation using the Gazebo [9] simulator. Gazebo provides significant realism as it operates in a three dimensional world with rigid body dynamics. The robot we used is a simulated Pioneer 2 DX mobile base with a laser range finder, monocular camera, and gripper. Figure 3 depicts the simulated environment for one of the exploration trials.

For the exploration task, the teacher's goal is to instruct the robot how to effectively explore an indoor environment. In order to properly validate our algorithm, we performed three separate experiments. The first two took place in a single-room environment. The first demonstration had the robot follow a counter-clockwise path around the room, and the second used a zig-zag pattern; the two paths are shown in Figure 1(a) and Figure 1(b). In both cases, the robot was instructed that the salient features are the distances to the nearest walls on the left and right. The third indoor exploration task took place in a more complex room, shown in Figure 1(c). The robot was told that the salient features are distances to nearest left, right, and front walls.

The pick-and-place task represents a complex teaching scenario in a domain other than exploration. The environment for this task contained two sets of colored boxes. Each set had two boxes colored red and blue. The goal of the task was to place similarly colored boxes together. The robot started in an arbitrary position, and the teacher commanded its navigation in the same manner as in the above-described exploration tasks. The teacher also indicated pick and place points/times. The salient features in this task are the distances to each of the colored objects.

IX. Experimental Results

For the first exploration task, as the robot followed the instructors commands two logs were created for the left and right wall distances. Only one segmentation point was found at the point when robot left the hallway, as shown in Figure 1(a). At this location, both left and right wall features reported a segmentation point. For the remaining path the only valid data consisted of the right wall distance feature, except when the robot passed around the circular object. During this time the left wall feature reported two segmentation point while the right wall feature reported none. These segmentation points were eliminated since they did not match between the two features.

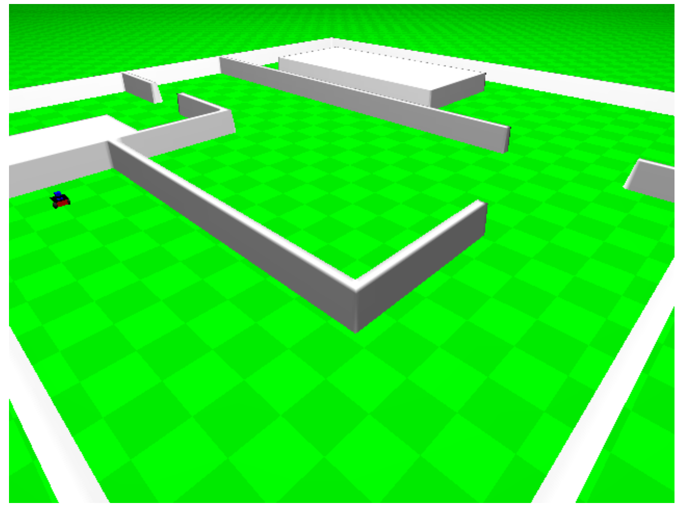


Fig. 3. Gazebo simulation environment of the multi-room exploration task. A Pioneer 2 DX robot is in the upper left at the demonstration starting point.

The second exploration task involved the same environment with a different exploration strategy. In this experiment multiple segmentation points were found due to the complex path. In Figure 1(b), segmentation point 1 correlates to the hallway exit; point 2 occurred when the robot stopped upon sensing a wall on its left side; point 3 resulted from sensing the wall again after turning. Each subsequent point can be tied to a specific event in the robot's perceptual history.

The third demonstration took place in a complex environment consisting of multiple rooms of varying sizes, as shown in Figure 1(c). Again, segmentation points occurred at times of significant change in the salient features.

For all three of the demonstrations, the task was decomposed into a set of simple, consistent behaviors. For example, in all the demonstrations the robot determined that the first portion of the path contained similar feature data correlating to what we would call a corridor-following behavior. Subsequent segments can be thought of as wall following behaviors and open-space traversals. It is interesting to note that behaviors repeat themselves, consistent with the notion of using a possibly small set of set of behaviors to accomplish a large number of tasks [11], [12].

The pick-and-place task involved navigating the robot to each colored object in a specific sequence. Figure 1(d) shows the path the robot followed as it first picked up a red object, dropped it off at the other red object, and then repeated the process for the blue object.

The salient features recorded during the pick-and-place task included the distance to each colored object. Referring to Figure 1(d), the segment [S,1] correlates to a behavior for approaching and taking the red object. In Segment [1,2], the robot did not see any colored object, and therefore had no feature information. In the following segment [2,3], the robot detected the second red object, approached it, and dropped off the first red object. The remaining segments are similar in nature. In Segment [3,4] there are no features data, [4,5] approach and pick up blue object, [5,6] no feature data, and [6,E] approach second blue object and drop off first.

During this demonstration the robot reached three separate points where it simultaneously detected two separate colored objects. These points occurred at the start of the demonstration, and twice while moving through the center of the environment. In each case, the result was a segmentation point that did not actually mark a logical boundary between behaviors. However, segmentation points occurred out of sync with other features and were transient. This resulted in the error cases being filtered out since they did not match temporally with another segmentation point.

X. Summary and Future Work

We have described a simple and efficient algorithm for automatic segmentation of feature sequences into distinct behaviors. The feature sequences in our work were generated through teaching demonstrations, as we aim to apply the segmentation algorithm as a means of enabling robot programming and reprogramming from demonstration. In both a simple task of room exploration and a more complex pick-and-place task, the algorithm was capable of determining the relevant segmentation points.

The next step involves using the segmented behaviors as a basis for new task learning and generalization. Ultimately, if teaching is done progressively, the robot will be capable of building a repertoire of behaviors applicable across of wide range of tasks, enabling natural learning and human-robot interaction.

References

[1] R. C. Arkin. *Behavior-Based Robotics*. MIT Press, CA, 1998.

[2] C. G. Atkenson and S. Schaal. Robot learning from demonstration. In Jr. Douglas H. Fisher, editor, *Machine Learning: Proceedings of the Fourteenth Internation Conference (ICML '97)*, pages pp. 12–20, San Francisco, CA, 1997. Morgan Kaufmann.

[3] Richard Bellman. On the approximation of curves by line segments using dynamic programming. *Commun. ACM*, 4(6):284, 1961.

[4] A. Billard, Y. Epars, S. Calinon, G. Cheng, and S. Schaal. Discovering optimal imitation strategies. *Robotics and Autonomous Systems*, 47:2-3, 2004.

[5] David P. Dobkin, Allan R. Wilks, Silvio V. F. Levy, and William P. Thurston. Contour tracing by piecewise linear approximations. *ACM Trans. Graph.*, 9(4):389–423, 1990.

[6] Xianping Ge and Padhraic Smyth. Deformable markov model templates for time-series pattern matching. In *KDD-2000 (Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining)*, pages 81–90, Boston, MA USA, Aug 2000.

[7] H. Imai and M. Iri. An optimal algorithm for approximating a piecewise linear function. *Journal of Information Processing*, 9(3):159–162, 1986.

[8] Eamonn J. Keogh, Selina Chu, David Hart, and Michael J. Pazzani. An online algorithm for segmenting time series. In *ICDM '01: Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 289–296, Washington, DC, USA, 2001. IEEE Computer Society.

[9] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2149–2154, Sendai, Japan, Sep 2004.

[10] J. Lin. Divergence measures based on the shannon entropy. *IEEE Transaction on Information Theory*, 37(1):145–151, 1991.

[11] Maja Matarić. Behavior base control: Examples from navigation, learning, and group behavior. *Experimental and Theroretical Artificial Intelligence*, 9(2-3):323–326, 1997.

[12] Maja Matarić and Matthew J. Marjanovic. Synthesizing complex behaviors by composing simple primitives. In *Self Organization and Life, From Simple Rules to Global Complexity, European Conference on Artificial Life (ECAL-93)*, pages 698–707, Brussels, Belgium, May 1993.

[13] S. Münch, J. Kreuziger, M. Kaiser, and R. Dillmann. Robot programming by demonstration (rpd) - using machine learning and user interaction methods for the development of easy and comfortable robot programming systems. In *25th International Symposium on Industrial Robots (ISIR '94)*, pages 685–693, Hanover, Germany, April 1994.

[14] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, and M. Kawato. Learning from demonstration and adaptation of biped locomotion. *Robotics and Autonomous Systems*, 47(2-3):79–81, 2004.

[15] Monica Nicolescu and Maja J. Matarić. Natural methods for robot task learning: Instructive demonstration, generalization and practice. In *International Joint Conference on Autonomous Agents and Multiagent Systems*, Melbourne, Australia, Jul 2003.

[16] Anand Panangadan, Maja J. Matarić, and Gaurav S. Sukhatme. Detecting anomalous human interactions using laser range-finders. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2136–2141. IEEE Press, Sep 2004.