Training a Robot via Human Feedback: A Case Study

W. Bradley Knox[1], Peter Stone[2], and Cynthia Breazeal[1]

[1] Massachusetts Institute of Technology
Media Lab, 20 Ames Street, Cambridge, MA USA
bradknox@mit.edu, cynthiab@media.mit.edu
[2] University of Texas at Austin
Dept. of Computer Science, Austin, TX USA
pstone@cs.utexas.edu

Abstract. We present a case study of applying a framework for learning from numeric human feedback—TAMER—to a physically embodied robot. In doing so, we also provide the first demonstration of the ability to train multiple behaviors by such feedback without algorithmic modifications and of a robot learning from free-form human-generated feedback without any further guidance or evaluative feedback. We describe transparency challenges specific to a physically embodied robot learning from human feedback and adjustments that address these challenges.

1 Introduction

As robots increasingly collaborate with people and otherwise operate in their vicinity, it will be crucial to develop methods that allow technically unskilled users to teach and customize behavior to their liking. In this paper we focus on teaching a robot by feedback signals of approval and disapproval generated by live human trainers, the technique of *interactive shaping*. These signals map to numeric values, which we call "human reward".

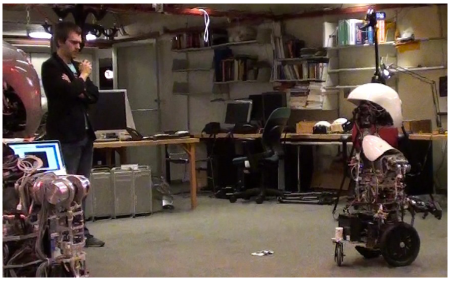

Fig. 1. A training session with the MDS robot Nexi. The artifact used for trainer interaction can be seen on the floor immediately behind Nexi, and the trainer holds a presentation remote by which reward is delivered.

In comparison to learning from demonstration [1], teaching by such feedback has several potential advantages. An agent can display its learned behavior while being taught by feedback, increasing responsiveness to teaching and ensuring that teaching is focused on the task states experienced when the agent behaves according to its learned policy. A feedback interface can be independent of the task domain. And we speculate that feedback requires less expertise than control and places less cognitive load on the trainer. Further, a reward signal is relatively simple in comparison a control signal; given this simplicity, teaching by human-generated reward is a promising technique for improving the effectiveness of low-bandwidth myolectric and EEG-based interfaces, which are being developed to enable handicapped users to control various robotic devices.

G. Herrmann et al. (Eds.): ICSR 2013, LNAI 8239, pp. 460–470, 2013.

In this paper, we reductively study the problem of learning from live human feedback in isolation, without the added advantages of learning from demonstration or similar methods, to increase our ability to draw insight about this specific style of teaching and learning. This paper demonstrates for the first time that TAMER [6]—a framework for learning from human reward—can be successfully applied on a physically embodied robot. We detail our application of TAMER to enable the training of interactive navigation behaviors on the Mobile-Dexterous-Social (MDS) robot "Nexi". Figure 1 shows a snapshot of a training session. In this domain, Nexi senses the relative location of an artifact that the trainer can move, and the robot chooses at intervals to turn left or right, to move forward, or to stay still. Specifically, the choice is dependent on the artifact's relative location and is made according to what the robot has learned from the feedback signals provided by the human trainer. The artifact can be moved by the human, permitting the task domain itself—not just training—to be interactive. The evaluation in this paper is limited with respect to who trains the agent (the first author). However, multiple target behaviors are trained, giving the evaluation a different dimension of breadth than previous TAMER experiments, which focused on the speed or effectiveness of training to maximize a predetermined performance metric [9,11,8].

Though a few past projects have considered this problem of learning from human reward [4,21,20,16,18,13,9], only two of these implemented their solution for a robotic agent. In one such project [13], the agent learned partially in simulation and from hard-coded reward, demonstrations, and human reward. In another [18], the human trainer, an author of that study, followed a predetermined algorithm of giving positive reward for desired actions and negative reward otherwise. This paper describes the first successful teaching of a robot purely by free-form human reward. One contribution of this paper is the description of how a system for learning from human reward—TAMER—was applied to a physically embodied robot. A second contribution is explicitly demonstrating that different behaviors can be trained by changing only the reward provided to the agent (and trainer interaction with its environment). Isbell et al. [4] showed the potential for such personalization by human reward in a virtual online environment, but it has not previously been demonstrated for robots or for TAMER.

2 Background on TAMER

TAMER (for Training an Agent Manually via Evaluative Reinforcement) is a solution to the problem of how an agent can learn to perform a sequential task given only real-valued feedback on its behavior from a human trainer. This problem is defined formally by Knox [6]. The human feedback—"human reward"—is delivered through push buttons, spoken word, or any other easy-to-learn interface. The human's feedback is the *only* source of feedback or evaluation that the agent receives. However, TAMER and other methods for learning from human reward can be useful even when other evaluative information is available, as has been shown previously [21,5,17,11]. The TAMER algorithm described below has additionally been extended to learn in continuous action spaces through an actor-critic algorithm [22] and to provide additional information to the trainer—either action confidence or summaries of past performance—creating changes in the quantity of reward instances given and in learned performance [14].

Motivation and Philosophy of TAMER. The TAMER framework is designed around two insights. First, when a human trainer evaluates some behavior, she considers the long-term impact of that behavior, so her feedback signal contains her full judgment of the desirability of the targeted behavior. Second, a human trainer's feedback is only delayed by how long it takes to make and then communicate an evaluation. TAMER assumes that trainers' feedback is focused on recent behavior; as a consequence, human reward is considered a trivially delayed, full judgment on the desirability of behavior. Following the insights above and TAMER's assumption of behavior-focused feedback, TAMER avoids the credit assignment problem inherent in reinforcement learning. It instead treats human reward as fully informative about the quality of recent actions from their corresponding states.

Mechanics of TAMER. The TAMER framework consists of three modules, as illustrated in Figure 2: credit assignment to create labels from delayed reward signals for training samples; supervised learning from those samples to model human reward; and action selection using the human reward model. The three modules are described below.

TAMER models a hypothetical human reward function, $R_H : S \times A \rightarrow \mathbb{R}$, that predicts numeric reward based on the current state and action values (and thus is Markovian). This modeling, corresponding to the "supervised learner" box in Figure 2, uses a regression algorithm chosen by the agent designer; we call the model $\hat{R}_H$. Learning samples for modeling are constructed from experienced state-action pairs and the real-valued human reward credited to each pair as outlined below.

The TAMER algorithm used in this paper (the "full" algorithm with "delay-weighted aggregate reward" described in detail by Knox [6]), addresses the small delay in providing feedback by spreading each human reward signal among multiple recent state-action pairs, contributing to the label of each pair's resultant sample for learning $\hat{R}_H$. These samples, each with a state-action pair as input and a post-assignment reward label as the output, are shown as the product of the "credit assigner" box in Figure 2. Each sample's share of a reward signal is calculated from an estimated probability density function for the delay in reward delivery, f_{delay}.

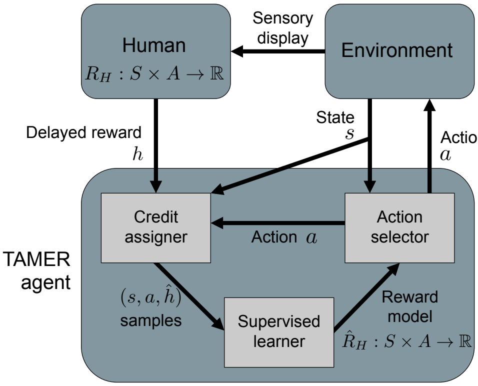

Fig. 2. An information-flow diagram illustrating the TAMER framework

To choose actions at some state s (the "action selector" box of Figure 2), a TAMER agent directly exploits the learned model $\hat{R}_H$ and its predictions of expected reward. When acting greedily, a TAMER agent chooses the action $a = argmax_a[\hat{R}_H(s, a)]$. This is equivalent to performing reinforcement with a discount factor of 0, where reward acquired from future actions is not considered in action selection (i.e., action selection is *myopic*). In practice, almost all TAMER agents thus far have been greedy, since the trainer can punish the agent to make it try something different, reducing the need for other forms of exploration.

Putting TAMER in Context. Although reinforcement learning was inspired by models of animal learning [19], it has seldom been applied to reward created from non-expert humans. We and others concerned with the problem of learning from human reward (sometimes called interactive shaping) seek to understand how reinforcement learning can be adapted to learn from reward generated by a live human trainer, a goal that may be critical to the usability of reinforcement learning by non-experts. TAMER, along with other work on interactive shaping, makes progress towards a second major form of teaching, one that will complement but not supplant learning from demonstration (LfD). In contrast to LfD, interactive shaping is a young approach. A recent survey of LfD *for robots* cites more than 100 papers [1]; this paper describes the second project to involve training of robots exclusively from human reward (and the first from purely *free-form* reward).

In comparison to past methods for learning from human reward, TAMER differs in three important ways: (1) TAMER addresses delays in human evaluation through credit assignment, (2) TAMER learns a model of human reward ($\hat{R}_H$), and (3) at each time step, TAMER *myopically* chooses the action that is predicted to directly elicit the maximum reward ($argmax_a \hat{R}_H(s, a)$), eschewing consideration of the action's effect on future state. Accordingly, other algorithms for learning from human reward [4,21,20,16,18,13] do not directly account for delay, do not model human reward explicitly, and are not fully myopic (i.e., they employ discount factors greater than 0).

However, nearly *all* previous approaches for learning from human-generated reward are *relatively* myopic, with abnormally high rates of discounting. Myopia creates certain limitations, including the need for the trainer to communicate what behavior is correct in any context (e.g., going left at a certain corner); a non-myopic algorithm instead would permit communication of correct outcomes (e.g., reaching a goal or failure state), lessening the communication load on trainers (while ideally still allowing behavior-based feedback, which people seem inclined to give). However, the myopic trend in past work was only recently identified and justified by Knox and Stone [10], who built upon this understanding to create the first successful algorithm to learn non-myopically from human reward [12]. Along with their success in a 30-state grid world, they also showed that their non-myopic approach needs further human-motivated improvements to scale to more complex tasks.

Complementing this continuing research into non-myopic approaches, this paper focuses on applying an established and widely successful myopic approach to a robotic task, showing that TAMER can be used flexibly to teach a range of behaviors and drawing lessons from its application. TAMER has been implemented successfully in a number of simulation domains commonly used in reinforcement learning research: mountain car [9], balancing cart pole [11], Tetris [9], 3 vs. 2 keep-away soccer [17], and a grid-world task [10]. In comparison, other interactive-shaping approaches have been applied in at most two domains.

3 The MDS Robot Nexi

A main contribution of this paper is the application of TAMER to a physical robot, shown in Figure 3a. The Mobile-Dexterous-Social robot platform is designed for research at the intersection of mobility, manipulation, and human-robot interaction [2]. The mobile base of the MDS platform has 2 degrees of freedom, with two powered wheels and one unpowered, stability-adding wheel. The robot estimates its environmental state through a Vicon Motion Capture system that determines the 3-dimensional locations and orientations of the robot and the training artifact; in estimating its own position and orientation, the robot employs both the Vicon data and information from its wheel encoders. In addition to the Vicon system, the robot has a number of other sensing capabilities that are not employed in this work.

4 TAMER Algorithm for Interactive Robot Navigation

We implemented the full TAMER algorithm as described generally in Section 2 and in detail by Knox [6], using the delay-weighted aggregate reward credit assignment system described therein.

From the robot's estimation of the position and orientation of itself and the training artifact, two features are extracted and used as input to $\hat{R}_H$ along with the action. The first feature is the distance in meters from the robot to the training artifact, and the second is the angle in radians from the robot's position and orientation to the artifact. Figure 3b shows these state features and the four possible actions: turn left, turn right, move forward, or stay still.

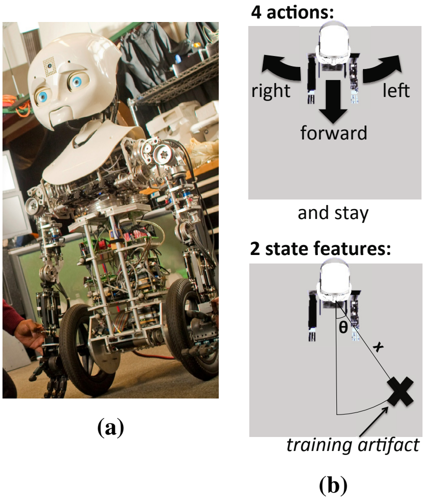

Fig. 3. **(a)** The MDS robot Nexi. **(b)** Nexi's action and state spaces, as presented to TAMER.

In implementing a version of TAMER that learns interactive navigational behaviors, we specified the following components. $\hat{R}_H$ is modeled by the k-nearest neighbors algorithm. More detail is given later in this section. The *training interface* is a presentation remote that can be held in the trainer's hand. Two buttons map to positive and negative reward, giving values of $+1$ and -1 respectively. Also, an additional button on the remote toggles the training mode on and off. When toggled on, TAMER chooses actions *and learns from feedback on those actions*; when off, TAMER does not learn further but does demonstrate learned behavior (see Knox [6] for details about toggling training). Another button both turns training off and forces the robot to stay still. This safety function is intended to avoid collisions with objects in the environment. The probability density function f_{delay}, which is used by TAMER's credit assignment module and describes the probability of a certain delay in feedback from the state-action pair it targets, is a Uniform(-0.8 seconds, -0.2 seconds) distribution, as has been employed in past work [6].[1]

The duration of time steps varies by the action chosen (for reasons discussed in Section 5). Moving forward and staying each last 1.5 seconds; turns occur for 2.5 seconds. When moving forward, Nexi attempts to move at 0.075 meters per second, and Nexi seeks to turn at 0.15 radians per second. Since changes in intended velocity—translational or rotational—require a period of acceleration, the degree of movement during a time step was affected by whether the same action had occurred in the previous time step.

$\hat{R}_H$ is modeled using k-nearest neighbors with a separate sub-model per action (i.e., there is no generalization between actions), as shown in Algorithm 1. The number of neighbors k is dynamically set to the floor of the square root of the number of samples gathered for the corresponding action, growing k with the sample size to counteract the lessening generalization caused by an increase in samples and to reduce the impact of any one experienced state-action pair, reducing potential erratic behavior caused by mistaken feedback. The distance metric is the Euclidean distance given the 2-dimensional feature vectors of the queried state and the neighboring sample's state. In calculating the distance, each vector element v is normalized within [0, 1] by $(v - v_{min})/(v_{max} - v_{min})$, where v_{max} and v_{min} are respectively the maximum and minimum values observed across training samples in the dimension of v.

To help prevent one or a few highly negative rewards during early learning from making Nexi avoid the targeted action completely, we bias $\hat{R}_H$ toward values of zero. This biasing is achieved by reducing the value of each neighbor by a factor determined by its distance d from the queried state, with larger distances resulting in larger reductions. The bias factor is calculated as the maximum of linear and hyperbolic decay functions as shown in line 11 of Algorithm 1.

Lastly, when multiple actions have the same predicted reward from the current state, such ties are broken by repeating the previous action. This approach lessens the number of action changes, which is intended to reduce early feedback error caused by ambiguously timed changes in actions (as discussed in Section 5). Accordingly, at the first time step, during which all actions are tied with a value of 0, a random action is chosen and repeated until non-zero feedback is given.

Algorithm 1. Inference by k-Nearest Neighbors

Given: Euclidean distance function d over state features
Input: Query q with state $q.s$ and action $q.a$, and a set of samples M_a for each action a. Each sample has state features, an action, and a reward label $\hat{h}$.

```
1: $k \leftarrow floor(\sqrt{|M_{q.a}|})$
2: if $k = 0$ then
3:     $\hat{R}_H(q.s, q.a) \leftarrow 0$
4: else
5:     $knn = \emptyset$
6:     $preds\_sum \leftarrow 0$
7:     for $i = 1$ to $k$ do
8:         $nn \leftarrow argmin_{m \in M_{q.a} \setminus knn} d(m, q)$
9:         $knn \leftarrow knn \cup \{nn\}$
10:        $dist \leftarrow d(nn, q)$
11:        $prediction_i \leftarrow nn.\hat{h} \times$
               $max(1 - (dist/2), 1/(1 + (5 \times dist)))$
12:        $preds\_sum \leftarrow preds\_sum + prediction_i$
13:    end for
14:    $\hat{R}_H(q.s, q.a) \leftarrow preds\_sum/k$
15: end if
16: return $\hat{R}_H(q.s, q.a)$
```

[1] Two minor credit-assignment parameters are not explained here but are nonetheless part of the full TAMER algorithm. For this instantiation, these are $\epsilon_p = 1$ and $c_{min} = 0.5$.

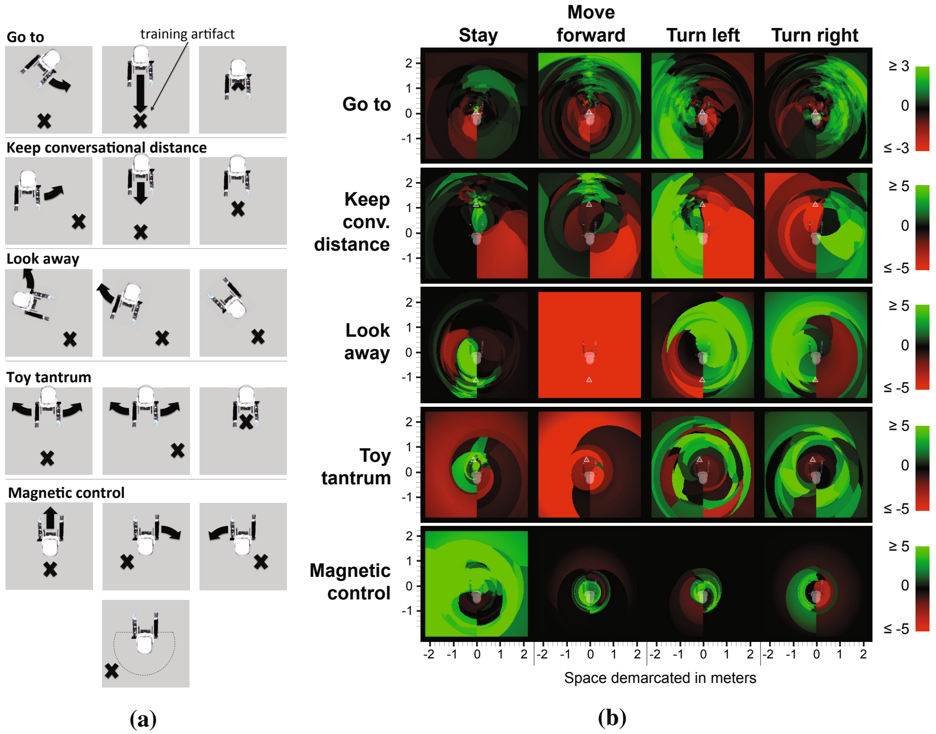

Fig. 4. (a) Iconic illustrations of the five interactive navigational behaviors that were taught to the MDS robot Nexi (described in Section 5). Each gray square represents a category of state space. The arrow indicates the desired action in such state; lack of an arrow corresponds to the stay action. **(b)** Heat maps showing the reward model that was learned at the end of each successful training session. Nexi is shown by a transparent birds-eye rendering of the robot, with Nexi facing the top of the page. The map colors communicate the value of the reward prediction for taking that action when the artifact is in the corresponding location relative to Nexi. A legend indicating the mapping between colors and prediction values for each behavior is given on the right. The small triangle, if visible, represents the location of the artifact at the end of training and subsequent testing of the behavior. (Note that in all cases the triangle is in a location that should make the robot stay still, a point of equilibrium.)

5 Results and Discussion

We now describe the results of training the robot and discuss challenges and lessons provided by implementing TAMER in this domain.

Behaviors Taught. Five different behaviors were independently taught by the first author, each of which is illustrated in Figure 4a:

- **Go to** – The robot turns to face the artifact, then moves forward, and stops before the artifact with little space between the two.
- **Keep conversational distance** – The robot goes to the artifact and stops at an approximate distance from the training artifact that two people would typically keep between each other during conversation (about 2 feet).

- **Look away** – The robot should turn away from the artifact, stopping when facing the opposite direction. The robot never moves forward.
- **Toy tantrum** – When the artifact is near the front of the robot, it does not move (as if the artifact is a toy that is in the robot's possession, satisfying the robot). Otherwise, the robot turns from side to side (as if in a tantrum to get the toy back). The robot never moves forward.
- **Magnetic control** – When the artifact is behind the robot, it acts as if the artifact repels it. The repulsion is akin to one end of a magnet repelling another magnet that faces it with the same pole. Specifically, when the artifact is near the center of the robot's back, the robot moves forward. If the artifact is behind its left shoulder, it turns right, moving that shoulder forward (and vice versa for the right shoulder). If the artifact is not near the robot's back, the robot does not move.

Videos of the successful training sessions—as well as some earlier, unsuccessful sessions—can be seen at `http://bradknox.net/nexi`. Figure 4b contains heat maps of the learned reward model for each behavior at the end of successful training.

Adjustments for Physical Embodiment. All of the videos were recorded during a one-day period of training and refinement of our implementation of the TAMER algorithm, during which we specifically adjusted action durations, the effects of chosen actions, and the communication of the robot's perceptions to the trainer. Nearly all sessions that ended unsuccessfully failed because of issues of *transparency*, which we addressed before or during this period. These transparency issues were mismatches between the state-action pair currently occurring and what the trainer believes to be occurring. The two main points of confusion and their solutions are described below.

The start and end of actions. As mentioned previously in Section 4, there can be a delay between the robot taking an action (e.g., turn right at 0.15 rad/s) and the robot visibly performing that action. This delay occurs specifically after any change in action. This offset between the robot's and the trainer's understandings of an action's duration (i.e., of a time step) can cause reward to be misattributed to the wrong action. The durations of each action—2.5 seconds for turns and 1.5 seconds otherwise—were chosen to ensure that the robot will carry out an action long enough that its visible duration can be targeted by the trainer.

The state of the training artifact. The position of the artifact, unlike that of the robot, was estimated from only Vicon data. When the artifact moved beyond the range of the infrared Vicon cameras, its position was no longer updated. The most common source of failed training sessions was a lack of awareness by the trainer of this loss of sensing. In response to this issue, an audible alarm was added that fired whenever the artifact could not be located, alerting the trainer that the robot's belief about the artifact is no longer changing.

The transparency issues above are illustrative of the types of challenges that are likely to occur with any physically embodied agent trained by human reward. Such issues are generally absent in simulation. In general, the designer of the learning environment and algorithm should seek to minimize cases in which the trainer gives feedback for a state-action pair that was perceived by the trainer but did not occur from the learning algorithm's perspective, causing misattributed feedback. Likewise, mismatches in perceived timing of state-action pairs could be problematic in any feedback-based learning system. Such challenges are related to but different from the correspondence problem in learning from demonstration [15,1], which is the problem of how to map from a demonstrator's state and action space to that of the emulating agent. Especially relevant is work by Crick et al. [3], which compares learning from human controllers who see a video feed of the robot's environment to learning from humans whose perceptions are matched to those of the robot, yielding a more limited sensory display. Their sensing-matched demonstrators performed worse at the task yet created learning samples that led to better performance.

Table 1. Training times for each behavior

Target behavior	**Active training time (in min.)**	**Total time (in min.)**
Go to	27.3	38.5
Keep conv. dist.	9.5	11.4
Look away	5.9	7.9
Toy tantrum	4.7	6.9
Magnetic control	7.3	16.4

The middle column shows the cumulative duration of active training time, and the right column shows the time of the entire session, including time when agent learning is disabled.

Training Observations. The *go to* behavior was taught successfully early on, after which the aforementioned state transparency issues temporarily blocked further success. After the out-of-range alarm was added, the remaining four behaviors were taught successfully in consecutive training sessions. Table 1 shows the times required for training each behavior. Note that the latter four behaviors—which differ from *go to* training in that they were taught using an anecdotally superior strategy (for space considerations, described at `http://bradknox.net/nexi`), had the alarm, and benefitted from an additional half day of trainer experience—were taught in considerably less time.

6 Conclusion

In this paper, we described an application of TAMER to teach a physically embodied robot five different interactive navigational tasks. The feasibility of training these five behaviors constitutes the first focused demonstration of the possibility of using human reward to flexibly teach multiple robotic behaviors, and of TAMER to do so in any task domain.

Further work with numerous trainers and other task domains will be critical to establishing the generality of our findings. Additionally, in preliminary work, we have adapted TAMER to permit feedback on intended actions [7], for which we plan to use Nexi's emotive capabilities to signal intention. One expected advantage of such an approach is that unwanted, even harmful actions can be given negative reward before they occur, allowing the agent to learn what actions to avoid without ever taking them. We are also developing methods for non-myopic learning from human reward, which will permit reward that describes higher-level features of the task (e.g., goals) rather than only correct or incorrect behavior, reducing the training burden on humans, permitting more complex behavior to be taught in shorter training sessions.

Acknowledgments. This work has taken place in the Personal Robots Group (PRG) at MIT and the Learning Agents Research Group (LARG) at UT Austin. PRG research is supported in part by NSF (award 1138986, Collaborative Research: Socially Assistive Robots). LARG research is supported in part by NSF (IIS-0917122), ONR (N00014-09-1-0658), and the FHWA (DTFH61-07-H-00030). We thank Siggi Örn, Nick DePalma, and Adam Setapen for their generous support in operating the MDS robot.

References

1. Argall, B., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstration. Robotics and Autonomous Systems 57(5), 469–483 (2009)
2. Breazeal, C., Siegel, M., Berlin, M., Gray, J., Grupen, R., Deegan, P., Weber, J., Narendran, K., McBean, J.: Mobile, dexterous, social robots for mobile manipulation and human-robot interaction. In: SIGGRAPH 2008: ACM SIGGRAPH 2008 New Tech Demos (2008)
3. Crick, C., Osentoski, S., Jay, G., Jenkins, O.C.: Human and robot perception in large-scale learning from demonstration. In: Proceedings of the 6th International Conference on Human-Robot Interaction, pp. 339–346. ACM (2011)
4. Isbell, C., Kearns, M., Singh, S., Shelton, C., Stone, P., Kormann, D.: Cobot in LambdaMOO: An Adaptive Social Statistics Agent. In: Proceedings of the 5th Annual International Conference on Autonomous Agents and Multiagent Systems (AAMAS) (2006)
5. Knox, W.B., Stone, P.: Combining manual feedback with subsequent MDP reward signals for reinforcement learning. In: Proceedings of the 9th Annual International Conference on Autonomous Agents and Multiagent Systems (AAMAS) (2010)
6. Knox, W.B.: Learning from Human-Generated Reward. PhD thesis, Department of Computer Science, The University of Texas at Austin (August 2012)
7. Knox, W.B., Breazeal, C., Stone, P.: Learning from feedback on actions past and intended. In: Proceedings of 7th ACM/IEEE International Conference on Human-Robot Interaction, Late-Breaking Reports Session (HRI 2012) (March 2012)
8. Knox, W.B., Glass, B.D., Love, B.C., Maddox, W.T., Stone, P.: How humans teach agents: A new experimental perspective. International Journal of Social Robotics, Special Issue on Robot Learning from Demonstration 4(4), 409–421 (2012)
9. Knox, W.B., Stone, P.: Interactively shaping agents via human reinforcement: The TAMER framework. In: The 5th International Conference on Knowledge Capture (September 2009)
10. Knox, W.B., Stone, P.: Reinforcement learning from human reward: Discounting in episodic tasks. In: 21st IEEE International Symposium on Robot and Human Interactive Communication (Ro-Man) (September 2012)
11. Knox, W.B., Stone, P.: Reinforcement learning with human and MDP reward. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS) (June 2012)
12. Knox, W.B., Stone, P.: Learning non-myopically from human-generated reward. In: International Conference on Intelligent User Interfaces (IUI) (March 2013)
13. León, A., Morales, E.F., Altamirano, L., Ruiz, J.R.: Teaching a robot to perform task through imitation and on-line feedback. In: San Martin, C., Kim, S.-W. (eds.) CIARP 2011. LNCS, vol. 7042, pp. 549–556. Springer, Heidelberg (2011)
14. Li, G., Hung, H., Whiteson, S., Knox, W.B.: Using informative behavior to increase engagement in the TAMER framework (May 2013)
15. Nehaniv, C.L., Dautenhahn, K.: 2 the correspondence problem. Imitation in Animals and Artifacts, 41 (2002)

16. Pilarski, P., Dawson, M., Degris, T., Fahimi, F., Carey, J., Sutton, R.: Online human training of a myoelectric prosthesis controller via actor-critic reinforcement learning. In: IEEE International Conference on Rehabilitation Robotics (ICORR), pp. 1–7. IEEE (2011)
17. Sridharan, M.: Augmented reinforcement learning for interaction with non-expert humans in agent domains. In: Proceedings of IEEE International Conference on Machine Learning Applications (2011)
18. Suay, H., Chernova, S.: Effect of human guidance and state space size on interactive reinforcement learning. In: 20th IEEE International Symposium on Robot and Human Interactive Communication (Ro-Man), pp. 1–6 (2011)
19. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. MIT Press (1998)
20. Tenorio-Gonzalez, A.C., Morales, E.F., Villaseñor-Pineda, L.: Dynamic reward shaping: Training a robot by voice. In: Kuri-Morales, A., Simari, G.R. (eds.) IBERAMIA 2010. LNCS, vol. 6433, pp. 483–492. Springer, Heidelberg (2010)
21. Thomaz, A., Breazeal, C.: Teachable robots: Understanding human teaching behavior to build more effective robot learners. Artificial Intelligence 172(6-7), 716–737 (2008)
22. Vien, N.A., Ertel, W.: Reinforcement learning combined with human feedback in continuous state and action spaces. In: 2012 IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL), pp. 1–6. IEEE (2012)