Robot Reinforcement Learning using Crowdsourced Rewards

Christian I. Penaloza[1], Sonia Chernova[2], Yasushi Mae[1] and Tatsuo Arai[1]

Abstract— **This work is inspired by the idea that robots should possess the ability to learn new skills with minimum guidance from a dedicated human teacher and they should enhance their abilities by utilizing external resources such as the Internet. In this paper we describe our approach for achieving robot learning by combining basic skill transfer through physical demonstrations, reinforcement learning and web-based crowdsource rewards obtained via remote laboratory. We demonstrate how a robot that is taught very basic motion skills through kinesthetic teaching (i.e. lift, push, etc.), can learn to manipulate several objects through reinforcement learning by taking advantage of the feedback from multiple non-expert crowd workers. Although reinforcement learning is commonly intractable for real robot applications due to high exploration times, we demonstrate that by transferring the training to the crowd, it is possible to achieve robot learning while minimizing the effort of a dedicated teacher. This paper describes the system architecture, as well as the integration of our application to a current remote laboratory framework (Robot Management System). Experimental results with a real robot performing a time consuming learning task show a successful learning performance using crowdsouce feedback.**

I. INTRODUCTION

Robots are rapidly integrating into human society and in the future they are expected to coexist with humans in every home [1]. However, most existing robots are pre-programmed to provide specific solutions and they lack of learning capabilities that would make them more adaptable to human environments. The ability to learn new skills is fundamentally important for robots to become more versatile and be able to perform everyday tasks.

Researchers have proposed diverse *Learning from Demonstration (LfD)* [2] methods to achieve robot learning for a variety of tasks. Most methods are based on supervised learning techniques that require the robot to learn from labeled data. The main idea behind these methods is to derive a policy that reproduces a demonstrated behavior from a set of example demonstrations performed by a human teacher. Example applications of robot learning from demonstration include object grasping [3], gesture learning [4] and full-body movement learning [5].

Although learning from demonstration techniques solve the problem of manual pre-programming of robots, the disadvantage is that human demonstrations might contain unclear data (due to differences of embodiment) which may affect the final policy performance learned by the robot. Moreover, teaching a robot to perform all sorts of different tasks is a tremendous duty that a human teacher might not enjoy. Ideally, a human could teach the robot basic skills, and the robot should learn independently to reuse those skills for different tasks.

Independent learning is usually achieved by Reinforcement Learning [6], in which an agent learns through exploration. While this technique has been successful in software agents and simulation applications [7], the large amount of data and high exploration times make it unpractical for real world robot applications. Recently, researchers have proposed interactive reinforcement learning techniques that combine human guidance and reinforcement learning, which improve the policy learning by reducing exploration times in both software agents [8]-[12] and real robots [13], [14]. Although interactive reinforcement learning seems to be an efficient learning method for real robots, the fact that a dedicated human teacher has to spend time guiding and rewarding the robot, makes it unpractical for future robot owners. For this reason, we believe that the robot should be able to use external resources such as web services to obtain the guidance or rewards needed to successfully learn the task.

Web-based crowdsourcing is a type of web service that has recently gained the attention of researchers to achieve robot control [15] and robot learning [16]-[19]. Crowdsourcing has a great potential to help robots learn new tasks because a small contribution of multiple crowd workers can sum up the total amount of guidance needed for the robot to learn the task without requiring a dedicated human teacher to be with the robot all the time. Crowdsourcing techniques have proved to be an effective approach where the crowd provides labeled data that the robot uses to learn tasks such as navigating in a maze [17], recognizing human actions [18] or grasping objects [19]. On the other hand, we believe that Reinforcement Learning techniques can also benefit from rewards provided by multiple crowd workers over the web.

In this work, we propose that robots should have the ability to: 1) learn basic skills from a human teacher, 2) use those skills to try to learn independently a task through exploration, and 3) take advantage of web-based crowdsourcing to obtain the guidance required to learn the task. Following this guideline, in this paper we demonstrate how a robot that is taught very basic motion skills through kinesthetic teaching (i.e. lift, push, etc.), can learn to manipulate several objects through Reinforcement Learning by taking advantage of the feedback from multiple non-expert crowd workers.

The remaining part of this work is organized as follows.

[1]C. Penaloza, Y. Mae and T. Arai are with the Graduate School of Engineering Science, Osaka University. 1-3 Machikaneyama-cho, Toyonaka, Osaka, Japan. penaloza,mae,arai at arai-lab.sys.es.osaka-u.ac.jp

[2]S. Chernova is with the Department of Computer Science, Worcester Polytechnic Institute, 125 Atwater Kent Labs, Worcester, MA, 01609. soniac at cs.wpi.edu

Section II presents the related literature. Section III describes the system architecture as well as the integration of our application to a current remote laboratory framework (Robot Management System). Experiment design and results are given in the subsequent section. Finally, in Section V we present some conclusive remarks and directions of future work.

II. RELATED WORK

Techniques that combine human guidance, whether in the form of demonstrations or human rewards, with reinforcement learning (RL) have a great potential for real world robot applications because on one side, human guidance leverages the experience and knowledge of the person teaching the robot, and on the other side, RL can be used for learning through exploration in complex domains. Example methods for combining human guidance with RL can be observed in the work of Taylor *et al.* [7], who developed *Human-Agent Transfer (HAT)* algorithm that combines LfD with RL; Knox *et al.* [11], who developed TAMER framework that learns to predict and maximize human rewards; Smart *et al.* [10], who developed a learning framework for mobile robots where robots first observe human demonstrations and then learn by RL; and Interactive Reinforcement Learning (Int-RL) techniques used in both software agents [8]-[12] and real robots [13], [14]. Although these techniques are good approaches for achieving robot learning, the fact that a human teacher has to be with the robot all the time represents an effort-time overload on behalf of the human teacher.

Recently, researchers have proposed novel methods to integrate web-based crowdsourced input into the robot learning loop. Most approaches are based on collecting labeled data generated by crowd workers while they perform a task. Chernova *et al.* [16], for example, used online games as a means of generating large-scale data for human-robot interaction research in order to create diverse interaction models. Crick *et al.* [17] collected data of multiple users during a robot teleoperation task in order to train a robot to navigate autonomously through a maze. Emeli [18] and Sorokin *et al.* [19] used a similar approach in which a robot collects data, uploads it to the cloud and crowd workers provide labels that are later used to learn to recognize human actions or grasp objects.

Although labeled data obtained from the crowd has been effective in aforementioned learning tasks, this might not be true for cases where crowd workers do not fully understand the task or they perform poorly. We believe that simplifying the task performed by the crowd to the level of just monitoring and evaluating robot actions, might benefit both workers and robots. For this reason, in this work we propose that crowd workers could only 1) observe a robot trying to learn a particular task using RL, and 2) provide positive and negative feedback through an intuitive interface.

Previous projects that involve crowdsourcing were performed using custom-made online platforms that only served the purpose of a particular experiment. However, with the recent development of robust remote laboratories, it is possible to achieve easier system integration and perform online experiments. While previous remote laboratories [22], [23] had attempted to foster shared research and experimentation, recent systems based on open source robot middleware ROS (Robot Operating System) [24], such as the PR2 Remote Lab [20], provide researchers with a more robust environment for remote development and experimentation on more capable robotics research platforms.

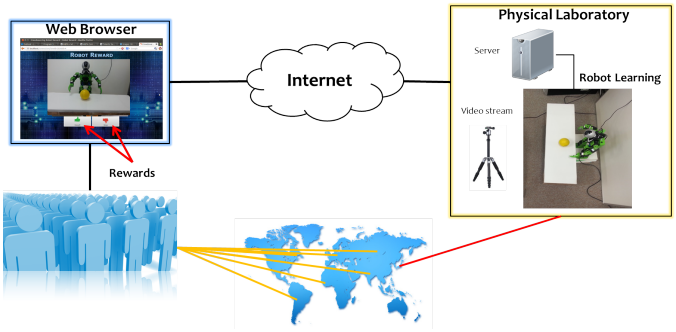


Fig. 1. Using RMS remote laboratory many people from around the world can observe a local robot experiment and provide feedback through a web browser

In this work, we used Robot Management System (RMS), a remote laboratory developed by Toris *et al.* [21] for the RobotsFor.Me (http://RobotsFor.me) project. RMS is built upon PR2 Remote Lab and is written in a robot-platform independent manner allowing the use of a variety of robots. By using RMS, many people from around the world can observe a local robot experiment and provide feedback through a web browser as described in Fig. 1. A key characteristic of RMS is that it was designed for use by the general public, and thus in the following sections we described how we integrated a LfD technique (kinesthetic teaching) with a reinforcement learning algorithm, and built an intuitive interface to collect crowdsource rewards using RMS.

III. SYSTEM ARCHITECURE

Building a remote laboratory that integrates LfD and RL in a local robot, along with a web-based interface for crowdsourcing experiments, might had been a tremendous task several years ago. However, with the availability of ROS and the aforementioned RMS, this task was reduced to building and integrating software modules that provide the functionality desired to perform online experiments that take effect in a local robot. This is because ROS provides a structured communications layer organized as a network of modules (*nodes*) that process data together, and RMS allows the easily installation, configuration and deployment of a stable web server that communicates effectively with different ROS nodes.

Our main contribution is the development and integration of 1) ROS nodes to achieve LfD through kinesthetic teaching, 2) ROS nodes to conduct RL based on Q learning algorithm, and 3) an intuitive user interface built upon RMS that collects crowd rewards. By using our system, we demonstrate how a robot that is taught very basic motion skills through

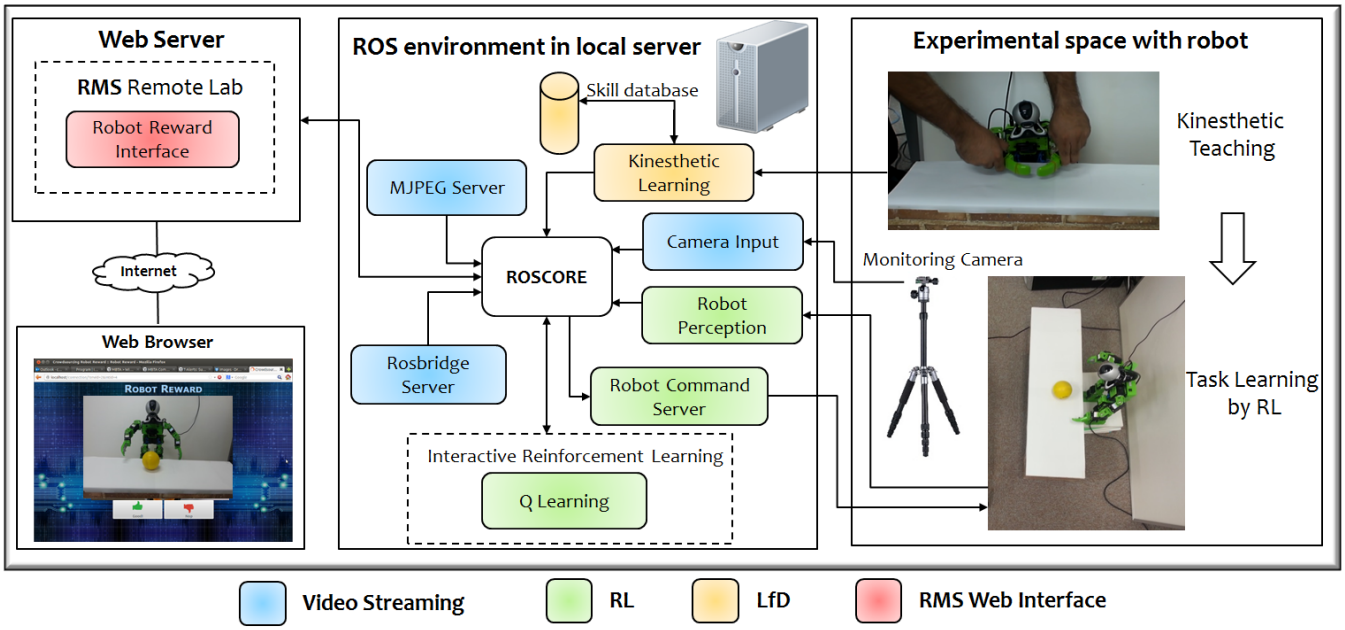


Fig. 2. Our system architecture features ROS nodes to achieve functionalities such as: LfD through kinesthetic teaching (orange), reinforcement learning (green), stream experiment video online (blue) and RMS web interface to display video and collect crowd rewards (red).

kinesthetic teaching (i.e. lift, push, etc.), can learn to manipulate several objects through RL by taking advantage of the feedback from multiple non-expert crowd workers.

Figure 2 shows a diagram that describes the interconnections among nodes in the system. The structured communications layer provided by ROS requires a particular node (*roscore*) to be running in order to allow other nodes to communicate. Nodes communicate by passing messages (*topics*) on publish/subscribe semantics in an asynchronous way, or through *services* in a synchronous way [24]. The following subsections describe the nodes involved in the different functionalities of the system, as well their interoperability.

A. Online Video Streaming of Local Experiment

Unlike crowdsourcing research in which crowd workers perform a particular task (tele-operation, game playing, etc), our approach only requires workers to observe a robot performing a task through a video stream, and provide feedback through button clicks. Video streaming was achieved by the following nodes provided by the ROS community:

- *Camera Input*[1] - acquires video feed from a physical camera and publishes the data to a topic.
- *MJPEG Server*[2] - subscribes to image topics and publishes those topics as MJPEG streams via HTTP.
- *Rosbridge Server*[3] - exposes data streams through web sockets accessible anywhere over the internet and provides run-time tools for interacting with and maintaining a robot system [24].

[1]http://www.ros.org/wiki/camera_drivers
[2]http://www.ros.org/wiki/mjpeg_server
[3]http://www.ros.org/wiki/rosbridge_suite

RMS communicates with ROS through Rosbridge Server in order to display the video and send data collected through a web browser.

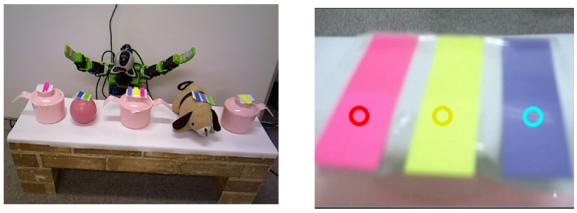
B. Integrating Kinesthetic Teaching

Our LfD approach consists in teaching the robot the skills needed to perform a task, rather than teaching the task itself. In this sense, we used kinesthetic teaching to demonstrate to a robot how to perform several actions with its arms (i.e. push with left arm, lift with both arms, etc.). Once the robot learned these skills, they became the basic building blocks that we used to teach the robot to manipulate different objects using RL (i.e. push ball with left arm, lift cup with both arms, etc.) through action exploration and rewards from the crowd.

In order to perform kinesthetic teaching, the teacher initiates a particular demonstration by moving the arms of the robot following the desired trajectory in a continuous manner. During the demonstration, the robot records joint angle trajectories of its arms, and when motion stops, collected data is saved as one demonstration. This process is repeated to give as many demonstrations as the teacher desires.

The robot used for our experiment is a 16DOF humanoid robot HOVIS Lite[4], in which we only considered 6DOF pertaining to robot's arms (3DOF each arm). After collecting 3 to 5 demonstrations for a particular action, we used Gaussian Mixture Models (GMM) to create a generalization of the trajectories as described in the work of Calinon *et al.* [25]. This learning takes place in the *Kinesthetic Learning* node of the system and all learned actions are saved in the *Skill database*. Finally the list of available skills is published in a topic where other nodes can subscribe to.

[4]http://www.dongburobot.com



(a) (b)

Fig. 3. Experiment setup: (a) Objects presented to the robot. (b) Each object had one or more color markers that represented the appropriate actions to manipulate the object.

C. Integrating Reinforcement Learning

The reinforcement learning algorithm used in our system is a modified version of the Q learning algorithm [27], in which we used crowdsourced rewards along with exploration to learn a task policy.

As a brief review, Q learning algorithm uses three elements, namely states, actions and rewards. The goal is to estimate the value of state-action pairs by generating a Q-table that is updated according to the rewards received after each iteration. More precisely, at every time step the robot observes its state $s \in S$ as a vector of k state variables such as $s = \langle s_1, s_2, \ldots, s_k \rangle$, selects an action from the set of available actions A, and receives a reward that may be positive or negative depending on the action selected for that particular state. Actions are chosen based on a policy $\pi : S \rightarrow A$ that is modified over time as the robot explores different actions and receives corresponding rewards. Using a reward function $R : S \times A \rightarrow \mathbb{R}$, the main objective is to maximize the total reward it receives after many iterations.

In our system, *Q Learning* node contains the core RL algorithm. As previously mentioned, Q learning requires the robot to observe its state and select an action in each iteration. In a real robot application, the state would correspond to a particular environmental or task state. *Robot Perception* node allows the robot to observe the state related to the particular learning task. In our experiment, *Robot Perception* node allows the robot to recognize color markers of different objects, as described more in detail in the subsequent section. On the other hand, a particular action is selected from the available actions stored in the *Skill database*, and sent through the *Robot Command Server* node to the robot to perform the physical action.

While the robot is performing a particular action, the RMS web interface collects the rewards in the form of click events performed by the crowd worker. Rewards are published in a topic through the Rosbridge Server, and the *Q Learning* node processes them using a majority vote scheme. In other words, when multiple crowd workers observe the same robot action, and provide different rewards (positive VS negative), only the most popular reward is assigned to that particular action-state.

D. Creating the User Interface in RMS

In order to make the task easier for crowd workers, we implemented a very simple interface that only has three components: video display, positive reward button (*like* icon) and negative reward button (*dislike* icon). Although the interface does not provide other means of interaction with the robot, such as tele-operation controls, it has the advantage that it is very intuitive and can easily be reused for different robot platforms.

Much of the infrastructure provided in RMS is in the form of widgets, which are modular stand-alone applications that can be easily added to the interface. A widget for video display was used in our interface, along with two HTML based buttons. The rosbridge server communicates with both: video display widget, to stream video data, and buttons, to receive click events and publish them to a topic.

TABLE I

STATE - ACTION

States	Action
s_1: Blue	*Push_w_LeftArm*
s_2: Green	*Push_w_RightArm*
s_3: Yellow	*Lift_w_LeftArm*
s_4: Pink	*Lift_w_RightArm*
s_5: Purple	*Lift_w_BothArms*

TABLE II

OBJECT - STATES

Object	States
Ball	s_1(blue), s_2(green)
Toy	s_1(blue), s_2(green), s_5(purple)
Left_handle_cup	s_3(yellow)
Right_handle_cup	s_4(pink)
Two_handle_cup	s_3(yellow), s_4(pink), s_5(purple)

IV. EXPERIMENT

In order to explore the benefits of crowdsourced feedback in a real-world robot, we designed a learning task that is representative of the kinds of tasks a user may one day want to teach his or her robot. Under traditional RL training conditions, this training process would be both monotonous and time-consuming for a single human teacher. We chose an object manipulation task in which the robot had to learn to manipulate objects. The robot chose its action through action exploration while crowd workers observed the experiment through RMS remote laboratory and provided feedback to the robot.

A. Learning Task

The experimental setup consisted of a robot standing behind a stationary table. A student volunteer used kinesthetic teaching to demonstrate to the robot how to perform five basic actions with one or two arms: *Push_w_LeftArm, Push_w_RightArm, Lift_w_LeftArm, Lift_w_RightArm* and *Lift_w_BothArms*. Objects with one or more color markers, shown in Fig. 3, were presented in front of the robot one at a time. Each color marker represented a state in the RL algorithm. Initially, the robot did not know which action corresponds to which state, and the goal of the experiment was to discover the state-action relationship shown in Table I. Each color marker can also be interpreted as an object property that defines the way a particular object can be manipulated. For example, an object (i.e. *Left_handle_cup*) that has a *yellow* property can be manipulated by the

Lift_w_LeftArm action. In this sense, objects may share one or more properties as shown in Table II.

Rather than having the robot learn one particular action for each object, we considered that learning the relationship of object properties-actions is more realistic, given the fact that a particular object can be manipulated in multiple ways.

Each learning trial represents an interaction with a single object. In every trial, the *Robot Perception* node of the system was used to recognize color markers of the object currently presented in front of the robot, as shown in Fig. 3b. If more than one color marker was recognized, only one color was randomly selected and assigned as the state to be evaluated by the RL algorithm. For a given state, one action was randomly selected from the available set of actions in the *skill database* and sent to the physical robot to execute. Crowdsourced rewards coming from RMS were collected, processed using majority vote and assigned to the state-action pair that was currently been evaluated. No change to the policy was made if the user did not provide a reward. Based on preliminary experiments, we found that approximately 300 episodes were needed to fully learn the task in an ideal scenario. To allow for sufficient time, each of the training sessions below lasted 60 minutes.

B. Crowdsourcing rewards

We performed three experiments with the objective of evaluating robot learning performance and the number of rewards provided by 1) a single crowd worker, 2) a group of crowd workers whose participation was pre-assigned in a sequential order, and 3) a group of crowd workers who participated in an unordered manner within the time assigned to the experiment.

A total of 25 participants: 14 female, 11 male (ages M=25.3, STD=3.98) participated in the experiment as crowd workers. They were randomly assigned to each of the three experiments that lasted 60 minutes. Before the day of the experiment, participants were given instructions about how login into RMS remote laboratory, what to expect during the experiment, and the way to provide positive/negative rewards to the robot according to its actions. The experimental conditions for each group are described as follows:

1) Experiment 1 - Rewards from a single person: (1) Only one participant was instructed to observe the robot learning task and provide rewards, (2) Participant could click several times to either positive or negative reward buttons, but only last click was considered, and (3) Participant was instructed that he could leave the experiment whenever he felt tired or bored.

2) Experiment 2 - Rewards from a crowd (sequential order): (1) 12 person participated, (2) Each participant was assigned a session of 5 minutes, and (3) Sessions were pre-assigned in sequential order.

3) Experiment 3 - Rewards from a crowd (unordered): (1) 12 persons participated, (2) Each person was asked to participate in a session of 5 to 10 minutes, and (3) Each participant was free to choose when to login within the assigned experiment time.

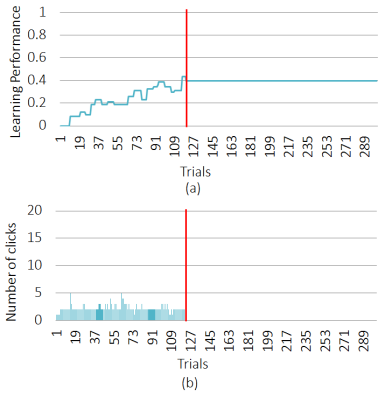


Fig. 4. Experiment 1: (a) unsuccessful learning performance (b) number of rewards/clicks. Red lines show the moment when the participant left the experiment.

C. Experimental Results

Figure 4 shows the results obtained in the first experiment where a single crowd worker observed the robot and provided rewards. The evolution of the learning performance (Fig. 4a) and the number of rewards/clicks (Fig. 4b) can be observed as the robot performed the trials. The participant was instructed that he could leave the experiment if he felt tired or bored; in this study, the participant left after 22.5 minutes, and this moment is marked as a red line in graphs (a) and (b). Learning performance stayed constant after participant stopped providing rewards and the maximum learning performance achieved was 41.86%. We believe that this unsuccessful learning performance may represent the case in which a human teacher stops guiding the robot in a monotonous task due to tiredness or boredom.

In experiments 2 (Fig. 5a) and 3 (Fig. 6a), in which the crowd trained the robot, the performance of the robot continued to improve, ultimately reaching a success rate of 93.12% and 91.56%, respectively after 60 minutes of continuous training. Fig. 5b and Fig. 6b show the cumulative number of reward clicks processed by the system during each trial. In the sequential condition, we found that single participants clicked between 1 and 5 times per trial; in the unordered condition, the number of rewards compared per trial in majority voting varied as participants logged in and out throughout the experiment.

Overall, our findings show that both crowdsourced conditions outperform the single user scenario. Although learning showed comparable levels of performance in the first 22 minutes of all three conditions (41.86%, 45.63% and 52.11%), a single user often does not have the time or patience required to perform all the training required. Our results suggest that the crowd can be leveraged to overcome this challenge. Interestingly, we see no significant difference between the two crowdsourcing conditions. We had hypothesized that majority voting among multiple inputs (experiment 3), would result in better performance as it would reduce the likelihood of the system receiving an incorrect reward from the user. However, our domain may have been too simple to see the effects of this factor.

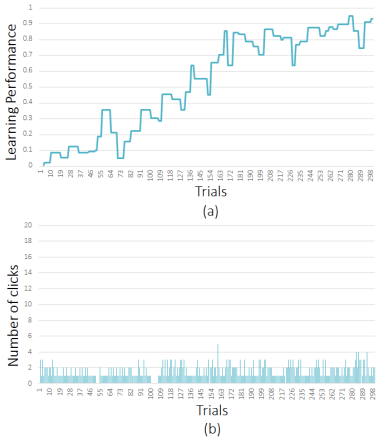


Fig. 5. Experiment 2: (a) Successful learning performance (b) number of rewards/clicks provided by participants in sequential order.

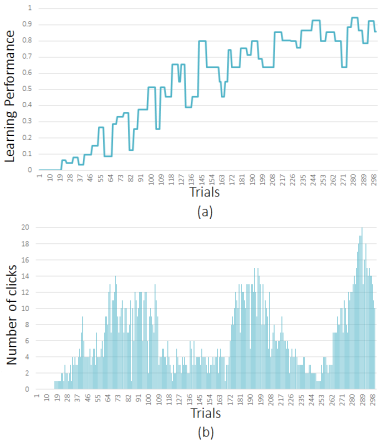


Fig. 6. Experiment 3: (a) Successful learning performance (b) number of rewards/clicks provided by participants in an unordered manner.

V. CONCLUSION

Robots require the ability to learn new skills with minimum guidance from human teachers. In this paper, we described our approach for achieving robot learning by combining basic skill transfer through physical demonstrations, reinforcement learning and web-based crowdsourced rewards obtained via remote laboratory. Experimental results demonstrate that by transferring the training to the crowd, it is possible to achieve a successful robot learning performance while minimizing the effort of a dedicated teacher. In future work, we will extend our framework to consider different mechanisms to take advantage of crowdsourced rewards for more complex and time-consuming learning tasks.

REFERENCES

[1] B. Gates. A robot in every home. Scientific American Magazine, 2006
[2] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. Robotics and Autonomous Systems, 469-483, 2009.
[3] J.D. Sweeney, R.A. Grupen, A model of shared grasp affordances from demonstration, in: Proceedings of the IEEE-RAS International Conference on Humanoids Robots, 2007
[4] S. Calinon, A. Billard, Incremental learning of gestures by imitation in a humanoid robot, in: Proceedings of the 2nd ACM/IEEE International Conference on Human-Robot Interactions, HRI'07, 2007.
[5] A. Ude, C.G. Atkeson, M. Riley, Programming full-body movements for humanoid robots by observation, Robotics and Autonomous Systems 47 (2004)
[6] Sutton, R. S., and Barto, A.G. 1998. Introduction to Reinforcement Learning. MIT Press.
[7] M.E. Taylor and P. Stone. Transfer learning for reinforcement learning domains: A survey. Journal of Machine Learning Research, 10(1):1633-1685, 2009.
[8] A. L. Thomaz and C. Breazeal, "Experiments in Socially Guided Exploration: Lessons learned in building robots that learn with and without human teachers.", Connection Science, Special Issue on Social Learning in Embodied Agents, 2008
[9] C. Breazeal and A. L. Thomaz. "Learning from human teachers with socially guided exploration.", In Proceedings of the International Conference on Robots and Automation, 2008
[10] Smart, W.D.; Kaelbling, L.P., "Effective reinforcement learning for mobile robots," Int. Conf. on Robotics and Automation. ICRA 2002
[11] W. B. Knox and P. Stone, Interactively shaping agents via human reinforcement: The TAMER framework, in The Fifth International Conference on Knowledge Capture, September 2009.
[12] A. L. Thomaz and C. Breazeal, Adding guidance to interactive reinforcement learning,h in In Proceedings of the Twentieth Conferencon Artificial Intelligence (AAAI), 2006.
[13] W. D. Smart, Making reinforcement learning work on real robots,Ph.D. dissertation, Department of Computer Science, Brown University,Providence, RI, 2002.
[14] Suay, H.B.; Chernova, S., "Effect of human guidance and state space size on Interactive Reinforcement Learning," RO-MAN, 2011 IEEE , vol., no., pp.1,6, 2011
[15] Walter S. Lasecki, Kyle I. Murray, Samuel White, Robert C. Miller and Jeffrey P. Bigham. Real-time Crowd Control of Existing Interfaces. In Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2011). Santa Barbara, CA, 2011.
[16] S. Chernova, N. DePalma, E. Morant, C. Breazeal, "Crowdsourcing human-robot interaction: Application from virtual to physical worlds", 20th IEEE International Symposium on Robot and Human Interactive Communication, 2011
[17] Christopher Crick, Sarah Osentoski, Graylin Jay and Odest Chadwicke Jenkins. Human and robot perception in large-scale learning from demonstration. Proceedings of the 6th ACM/IEEE International Conference on Human-Robot Interaction, 2011.
[18] Emeli, V., "Robot learning through social media crowdsourcing," Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on , vol., no., pp.2332,2337, 7-12 Oct. 2012.
[19] Sorokin A., Berenson D., Srinivasa S., and Hebert M., "People Helping Robots Helping People: Crowdsourcing for Grasping Novel Objects," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '10), October, 2010.
[20] Osentoski S., Pitzer B., Crick C., Graylin J., Dong S., Grollman D., Suay H.B., and Jenkins O.C. Remote Robotic Laboratories for Learning from Demonstration. In International Journal of Social Robotics, special issue on Learning from Demonstration, Volume 4, Issue 4, Page 449-461, 2012.
[21] Russell Toris and Sonia Chernova. RobotsFor.Me and Robots For You. Interactive Machine Learning Workshop, Intelligent User Interfaces Conference, 2013.
[22] Fu, M., Yeo, C., Lin, Y., Wang, F.: Waves: towards real time laboratory experiments in cyberspace. In: IEEE International Conference on Systems, Man, and Cybernetics, vol. 5, pp. 3470 3474 (2001)
[23] Callaghan, M., Harkin, J., Prasad, G., McGinnity, T., Maguire, L.: Integrated architecture for remote experimentation. In: IEEE International Conference on Systems, Man and Cybernetics, 2003.
[24] Quigley, M., Conley, K., Gerkey, B.P., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: Ros: an open-source robot operating system. In: ICRA Workshop on Open Source Software (2009)
[25] Crick, C., Jay, G., Osentoski, S., Pitzer, B., Jenkins, O.C.: Rosbridge: Ros for non-ros users. In: Proceedings of the 15th International Symposium on Robotics Research (ISRR) (2011)
[26] S. Calinon, F. Guenter, and A. Billard. On learning, representing and generalizing a task in a humanoid robot. IEEE Transactions on Systems, Man and Cybernetics, Part B. Special issue on robot learning by observation, demonstration and imitation, 2007.
[27] C. J. C. H. Watkins and P. Dayan, "Q-Learning", Machine Learning Journal, volume 8, number 3/4, Special Issue on Reinforcement Learning, 1992