Teaching Multi-Robot Coordination using Demonstration of Communication and State Sharing

(Short Paper)

Sonia Chernova and Manuela Veloso
Carnegie Mellon University
Pittsburgh, PA, USA
{soniac,veloso}@cs.cmu.edu

ABSTRACT

Solutions to complex tasks often require the cooperation of multiple robots, however, developing multi-robot policies can present many challenges. In this work, we introduce teaching by demonstration in the context of multi-robot tasks, enabling a single teacher to instruct multiple robots to work together through a demonstration of the desired behavior. Within this framework, we contribute two approaches for teaching coordination based on different communication and information sharing strategies. To enable the teacher to divide attention between multiple robots, each robot uses a confidence-based algorithm that allows it to regulate its autonomy and determine the need for demonstration. Evaluation is performed using two Sony QRIO robots learning a real-world collaborative ball sorting task.

Categories and Subject Descriptors

I.2.9 [**Artificial Intelligence**]: Robotics

General Terms

Algorithms, Performance

Keywords

Learning from demonstration

1. INTRODUCTION

Teaching by demonstration is a learning approach based on human-robot interaction that provides an intuitive interface for robot programming. Using this approach, a robot learns to imitate the behavior of a teacher by observing a demonstration of the task.

In the standard formalization of demonstration learning, a single robot is taught by a single teacher [2, 4, 5]. However, solutions to complex tasks often require the cooperation of multiple robots. In this work, we explore teaching by demonstration in the context of multi-robot tasks, enabling a single person to teach multiple robots to work together.

Multi-robot coordination has been extensively studied in robotics research and presents many challenges, such as issues of action coordination, communication, and physical interaction. In this paper, we focus on the teaching element of the multi-robot demonstration problem and contribute two approaches for teaching coordination. The first approach, *coordination through active communication*, enables the teacher to explicitly teach *when* information should be shared through demonstration of communication actions. The second approach, *coordination through shared state*, enables the teacher to select state information to be shared automatically between robots and to demonstrate only the physical actions to be performed.

When instructing multiple robots, the teacher is unable to interact with all robots at the same time and must shift attention between them. Each individual robot must therefore possess a degree of autonomy that makes it robust to intermittent periods of neglect from the teacher. To address this problem, we build upon our single-robot demonstration learning algorithm, Confident Execution [1]; this algorithm provides a decision-making mechanism that prevents the robot from acting autonomously in unknown or uncertain situations by actively selecting between autonomous execution and requests for demonstration. Using this mechanism, each robot no longer requires the teacher's undivided attention, allowing the teacher to work with multiple robots at the same time. We evaluate the multi-robot learning algorithm and coordination strategies by teaching two Sony QRIO humanoid robots to perform a ball sorting task.

2. MULTI-ROBOT LEARNING AND COORDINATION

One of the greatest challenges of extending demonstration learning to multi-robot systems is the problem of limited human attention, the fact that the teacher is not able to pay attention to, and interact with, all robots at the same time. Each individual robot must therefore possess a degree of autonomy that makes it robust to periods of neglect from the teacher. This challenge has prevented most existing single-robot algorithms from generalizing to multi-robot domains.

To address the problem of limited teacher attention we utilize the Confident Execution algorithm, a decision-making process that enables a robot to regulate its autonomy and request teacher demonstrations in unfamiliar states. Using this approach, each robot controls learning by selecting its own training data, requiring only intermittent teacher attention. In the following sections, we present an overview of Confident Execution in the context of multi-robot learning,

Cite as: Teaching Multi-Robot Coordination using Demonstration of Communication and State Sharing (Short Paper), Sonia Chernova and Manuela Veloso, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16.,2008, Estoril, Portugal, pp. 1183-1186.

Algorithm 1 Multi-Robot Confident Execution

```
1: for each robot do
2:     s ← GetState()
3:     (a_p, c) ← ClassifyState(s)
4:     if DemonstrationRequired(c) then
5:         a_d ← GetDemonstration()
6:         UpdatePolicy(s, a_d)
7:         ExecuteAction(a_d)
8:     else
9:         ExecuteAction(a_p)
```

before presenting two approaches to teaching multi-robot coordination through demonstration.

2.1 Confident Execution Algorithm

Algorithm 1 presents a pseudocode summary of the Confident Execution algorithm, using which an independent policy is learned for each robot. Confident Execution is an interactive learning algorithm in which the robot must select demonstration examples, in real time, as it interacts with the environment. At each decision timestep, the algorithm selects between demonstration and autonomy based on a measure of action-selection confidence. Specifically, given the current state of the robot, the algorithm queries a classifier representing the robot's policy, and obtains a policy action a_p and its classification confidence c (line 3). The classification confidence is then used to identify states in which additional demonstration will provide useful information and improve the robot's policy, as described in [1].

When demonstration is required, the robot requests help from the teacher and waits for a demonstration of the correct action. The demonstrated action, a_d, is then used to update the robot's policy before being executed by the robot (lines 6-7). If the robot is confident in its ability to select the correct action, however, a demonstration is not required, and the robot autonomously executes the policy-selected action a_p (line 9). The action selection process is repeated once action execution is complete.

Confident Execution guides the robot to incrementally acquire datapoints representing the desired behavior. As more datapoints are acquired over time, fewer novel states are encountered, and the autonomy of the robot increases. Task learning is complete once the robot is able to repeatedly perform the task correctly without requesting demonstrations. In the context of multi-robot systems, we take advantage of the partial robot autonomy afforded by the Confident Execution algorithm to enable the teacher to instruct multiple robots. Multi-robot coordination emerges from the interaction between robots based on their independent policies.

2.2 Demonstration of Multi-Robot Coordination

In addition to teaching multiple robots at the same time, we are interested in teaching them to work together. Coordination between individual robots relies on a common understanding of the world based on communicated information. In this paper, we contribute two techniques for teaching coordination using communication.

The ***coordination through active communication*** approach enables the teacher to use demonstration to explicitly teach when information should be communicated. Each robot's abilities are extended to include communication actions which are used to share locally observed state with the robot's teammates. During demonstration, the teacher selects among a set of both physical and communication actions for the robot to perform. Based on these demonstrations, communication actions are incorporated directly into the robot's action policy.

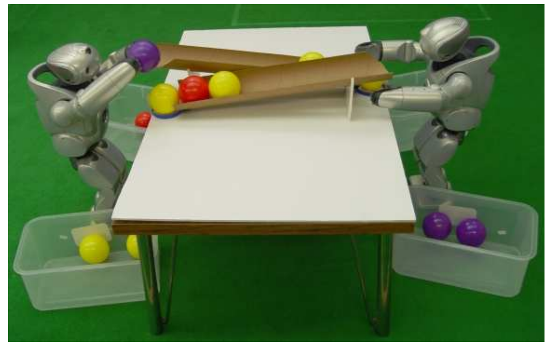

Figure 1: QRIO robots performing ball sorting task.

Coordination through shared state takes a different approach by automating the inter-robot communication process. This technique enables the teacher to select the locally observed state features each robot shares with its teammates. The algorithm then tracks the status of these features, and automatically communicates their values each time they change. During learning, the teacher focuses on demonstrating only the physical actions to be performed based on the shared state information. Note that this approach is targeted at discrete valued features that do not change rapidly over time. The evaluation of both teaching methods is presented in Section 4.

3. EXPERIMENTAL SETUP

Experimental evaluation of the presented algorithms was conducted using the Sony QRIO humanoid robots, Figure 1. The QRIO robot is a fully autonomous system enabled with 38 degrees of freedom, onboard processing, stereo vision and speech [3]. Wireless communication allows additional off-board processing to be integrated seamlessly into the system. The robot's anthropomorphic design and ability to express emotions through human-like motion and speech make it highly suitable for human-robot interaction.

3.1 Ball Sorting Domain

Evaluation of multi-robot learning and coordination was performed in a ball sorting domain. Figure 1 shows the robots operating in the domain, which consists of two sorting stations connected by ramps. Each station has an individual queue of colored balls (red, yellow or blue) that arrive via a sloped ramp for sorting. The robots' task is to sort the balls by color into four bins.

For this task, the action set of each robot consists of the following actions: *wait*, *sort left*, *sort right* and *pass*. The sorting actions enable the robot to pick up the ball at the head of its queue and place it into the left or right bin. Passing enables the robot to place the ball onto its teammate's ramp, where it rolls down to the tail end of the other robot's queue. The color and location of the balls is determined by each robot using its onboard vision system. Using these abilities, the robot are taught to perform the following task:

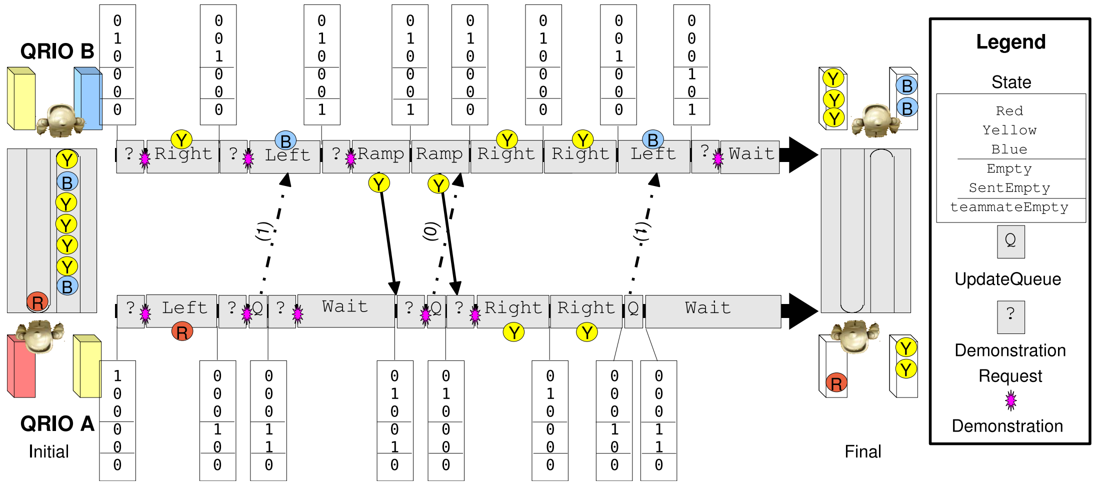

Figure 2: Ball sorting task execution using coordination through active communication.

Ball Sorting Task: Each robot begins with multiple balls of various colors in its queue. QRIO A sorts red and yellow balls into the left and right bins, respectively, and passes blue balls to QRIO B. QRIO B sorts blue and yellow balls into the left and right bins, respectively, and passes red balls to QRIO A. Additionally, we would like the number of sorted balls to be distributed fairly between the robots. If one robot runs out of balls, the other robot should pass additional balls into its teammate's queue. Only balls that can be sorted by the other robot should be passed, however. For example, QRIO A should pass only the blue and yellow balls, and QRIO B should pass only the red and yellow balls. If both queues are empty, the robots should wait.

3.2 Human-Robot Interaction

Learning from demonstration is an interactive process that requires two-way communication between the robot and the teacher. When requesting a demonstration, the QRIO robot attempts to audibly attract the teacher's attention by speaking the phrase "What should I do now?". Visually, uncertainty is indicated by an open arm gesture and the lighting of LEDs on the head.

The teacher interacts with each robot via a GUI interface, which allows him or her to select the action to demonstrate from among the available action primitives. The GUI additionally displays the robot's current state vector.

4. EVALUATION

In this section we evaluate and compare the active communication and shared state approaches to teaching multi-robot coordination. Each robot's locally observed state is represented by the boolean state vector $\{red, yellow, blue\}$, in which each feature represents the absence or presence of a ball of a particular color. Only the ball at the head of the queue can be observed by the robot, resulting in a single color value being set at any one time.

To perform the ball sorting task, each robot must communicate a single bit of information to its teammate – whether its queue is empty or full. This information is stored in the boolean state feature *Empty*, which is set to 0 if a ball is present in the queue. Finally, for each robot, the feature *teammateEmpty* represents its teammate's queue status.

4.1 Coordination through Active Communication

Communicating the values of *Empty* between robots at every timestep is typically impractical due to possible communication costs and network traffic congestion. Instead, we would like to use demonstration to teach the robot when communication should take place. To enable the teacher to explicitly demonstrate communication, we introduce a communication action, *UpdateQueueStatus*, which communicates the current status of a robot's queue. Our goal is for the robot to learn to perform this action each time the value of the *Empty* feature changes.

While most physical actions have an observable effect that changes the robot's state (i.e. moving an object changes the state of the environment), the immediate effect of communication can not be observed. For example, the robot can not sense that following the execution of *UpdateQueueStatus* the value of its teammate's *teammateEmpty* feature changes. To prevent the robot from remaining in the same state following a communication action, the state feature *SentEmpty* is added to represent the last communicated value of the *Empty* feature. The execution of *UpdateQueueStatus* causes both the local *SentEmpty* value and the teammate's state feature to be updated. A mismatch in the values of *Empty* and *SentEmpty* is an indication that an update is required.

In summary, the complete task setup is described by:

Robot State: $S = observed \cup sent \cup received$
$observed = \{red, yellow, blue\}$
$sent = \{Empty, SentEmpty\}$
$received = \{teammateEmpty\}$

Robot Actions: $A = physical \cup communication$
$physical = \{sort\ left, sort\ right, pass\ ramp, wait\}$
$communication = \{UpdateQueueStatus\}$

Figure 2 presents side-by-side timelines of the robots learning to perform the ball sorting task using active communication. Robot state at each decision point is represented by a vector of boolean values composed of the observed, sent and received features.

The initial robot configuration, in which QRIO B has seven balls and QRIO A has one, is shown on the left side of the figure. Both robots begin with no initial knowledge

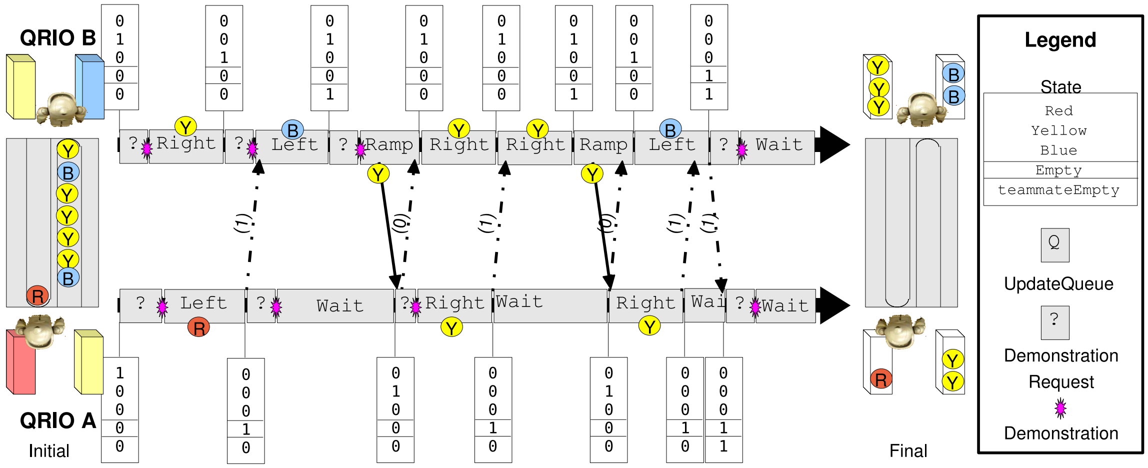

Figure 3: Ball sorting task execution using coordination through shared state.

about the task, and request a demonstration upon encountering the first state. The teacher instructs each robot to place its ball, red and yellow, into the left and right bin, respectively. Upon receiving their individual instructions, each robot executes the specified action.

At each following timestep, the robots evaluate their state, compare it to previously encountered demonstrations, and select between autonomous execution and demonstration. The boolean features used to represent the robot's state allow the classifier to learn quickly, requiring only a single demonstration per unique state. After sorting the red ball and emptying the queue, QRIO A asks for a second demonstration from the teacher. The teacher selects the *UpdateQueueStatus* action, which notifies QRIO B that its teammate ran out of balls.

As task execution continues, both robots begin to encounter repeated states and perform an increasing number of actions autonomously. The final configuration, in which QRIO A has gained two yellow balls, is shown on the right. Note that this example does not cover all possible robot states; teaching the entire task consisting of all possible combinations of balls and messages requires 32 demonstrations, 16 per robot.

4.2 Coordination through Shared State

Coordination based on communicated information relies on that information being up to date. Each time an important local change occurs, the robot must communicate its updated state to its teammates. Coordination through shared state automates this common communication case.

In the place of explicit communication actions used in the previous section, this approach utilizes a set of *shared state features*. For the ball sorting task, we select to share the state feature *Empty*, resulting in the following task configuration:

Robot State: $S = observed \cup shared \cup received$
$shared = \{Empty\}$
$received = \{teammateQueueEmpty\}$

Robot Actions: $A = physical$

Figure 3 presents a timeline of Task 2 being taught using shared state. Note that communication of the *Empty* feature, represented by dotted arrows, occurs automatically without the need for teacher demonstration. Using this approach, the same final state is reached by the robot, although the action order differs slightly due to fewer demonstration delays. The complete task requires 16 demonstrations, 8 per robot, to learn.

5. CONCLUSION

In this paper we presented an algorithm for teaching distributed multi-robot tasks through demonstration. Based on this algorithm, we additionally contributed two techniques for using demonstration to teach multi-robot coordination. The coordination through shared state approach focuses on automating the most common communication case in which shared information is always maintained up to date. Coordination through active communication presents a more general approach in which the teacher is able to encode any condition for communication into the robot's policy. Evaluation in a robotic ball sorting domain showed that both techniques can be successfully used to teach robots to coordinate. A promising direction for future work is to examine how these techniques can be combined to provide the most general solution to teaching multi-robot coordination.

6. REFERENCES

[1] S. Chernova and M. Veloso. Confidence-based policy learning from demonstration using gaussian mixture models. In *Proceedings of Autonomous Agents and Multiagent Systems*, May 2007.

[2] D. Grollman and O. Jenkins. Dogged learning for robots. In *IEEE International Conference on Robotics and Automation*, pages 2483–2488, 2007.

[3] T. Ishida. Development of a small biped entertainment robot QRIO. In *Micro-Nanomechatronics and Human Science, 2004*, pages 23–28, 2004.

[4] M. N. Nicolescu and M. J. Mataric. Natural methods for robot task learning: instructive demonstrations, generalization and practice. In *Proceedings of Autonomous Agents and Multiagent Systems*, pages 241–248. ACM Press, 2003.

[5] A. L. Thomaz. *Socially Guided Machine Learning*. PhD thesis, Massachusetts Institute of Technology, 2006.