Learning to Avoid Collisions

Elizabeth Sklar[1,4], **Simon Parsons**[1,4], **Susan L. Epstein**[2,4], **A. Tuna Özgelen**[4],
J. Pablo Muñoz[4], **Farah Abbasi**[3], **Eric Schneider**[2] and **Michael Costantino**[3]
[1]Brooklyn College, [2]Hunter College, [3]College of Staten Island, [4]The Graduate Center
The City University of New York
New York, NY USA
contact author: sklar@sci.brooklyn.cuny.edu

Abstract

Members of a multi-robot team, operating within close quarters, need to avoid crashing into each other. Simple collision avoidance methods can be used to prevent such collisions, typically by computing the distance to other robots and stopping, perhaps moving away, when this distance falls below a fixed threshold. While a simple method like this may skirt disaster, the results may be inefficient in terms of the amount of time that robots are halted, waiting for others to pass by, or in terms of the path traversed, moving around other robots. The experiments reported here describe a method in which a human operator, through a graphical user interface, watches robots performing an exploration task and can manually interrupt robots' movements before they crash into each other, and then resume their movements when their paths are clear. Experiment logs record the robots' state when they are halted and resumed, and a behavior pattern for collision avoidance is learned, by classifying the states in which "halt" and "resume" commands are issued. Preliminary work is reported here.

Introduction

We are interested in the use of human-robot teams to solve problems that are dangerous for entiely human teams to tackle, but are beyind the capabilities of entirely robot teams. As canonical examples of the kinds of task that fit this description we consider *urban search and rescue* (USAR)(Jacoff, Messina, and Evans 2000; Murphy, Casper, and Micire 2001) and *humanitarian de-mining* (Habib 2007; Santana, Barata, and Correia 2007). In urban search and rescue, robots explore an enclosed space, such as a collapsed building, and seek to locate human victims. In humanitarian de-mining, robots explore an open space, such as a field in a war zone, to search for anti-personnel mines that may be hidden from view. The goal is to locate mines so that they can be disarmed and the region rendered safe.

In both cases, teams of robots are deployed to locate targets of interest in terrain that is potentially unsafe for humans, and in both cases the robots will typically need a human operator to help with parts of the task that they cannot easily handle on their own. In the case of urban search and rescue this might be identifying a human victim, in the case of humanitarian demining, this might be determining what kind of device the robot team has located.

In our work (Sklar et al. 2011; 2012), we are especially interested in the use of inexpensive, limited function robots since we believe that teams of such robots are closer to deployment on the kinds of task we are interested in than teams made up of more expensive and more capable robots. Such robots have a particular need for human assistance, and there are a number of ways in which such robots can profitably learn from a human teacher. This paper reports on a preliminary investigation of one such instance, where the human operator trains the robot team members to avoid crashing into each other.

Related Work

The idea that an interactive system can improve its behavior through observation of human users' key strokes and mouse clicks, i.e., *data mining the clickstream*, is not new. In the 1960s and 70s, Teitelman developed an automatic error correction facility that grew into DWIM (Do What I Mean) (Teitelman 1979). In the early 1990's, Cypher created Eager, an agent that learned to recognize repetitive tasks in an email application and take over from the user (Cypher 1991). Maes used machine learning techniques to train agents to help with email, filter news messages, and recommend entertainment. These agents gradually gained confidence in their understanding of users' preferences (Maes 1994).

In robotics, the idea that robots can learn from humans has been explored in the area of *learning by demonstration* (**?**), also known as *programming by demonstration* (**?**). This is commonly viewed in the framework of reinforcement learning, with the focus being on learning a policy from a series of state/action pairs (**?**). Other approaches to robots learning from people include (**?**), where human teachers provide examples that seed evolutionary learning, (**?**), where the robot tries to identify the goal that a human is working towards in order to make its own plan to achieve the goal, and (**?**), where the robot observes the human carrying out actions in its domain and learns the outcomes of its own actions from these observations.

Little of this work is concerned with multiple robots. There is a long history of multi-robot learning, for example (**?**; **?**; **?**; **?**), but this is learning from trial and error, not learning from a human teacher.

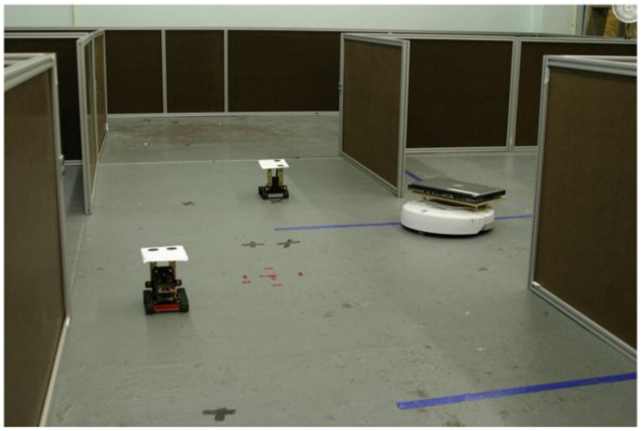


Figure 1: The robots' physical environment.

In earlier related work, we trained neural networks as controllers for emulator agents that play video and educational games, based on human moves collected during game play (Sklar 2000; Sklar, Blair, and Pollack 2001). The aim was not to produce the best player, but rather to derive a population of players that represent different characteristics of play. Sklar has trained agents to learn from self-play (Blair and Sklar 1999), from play against people (Funes et al. 1988; Sklar, Blair, and Pollack 2001; Sklar 2000) and from play against other agents (Blair, Sklar, and Funes 1998). More recently, she has extended the same technique beyond gaming to generate populations of agents that emulate students performing at different skill levels on an educational assessment (Sklar and Icke 2009; Sklar et al. 2007).

Our Approach

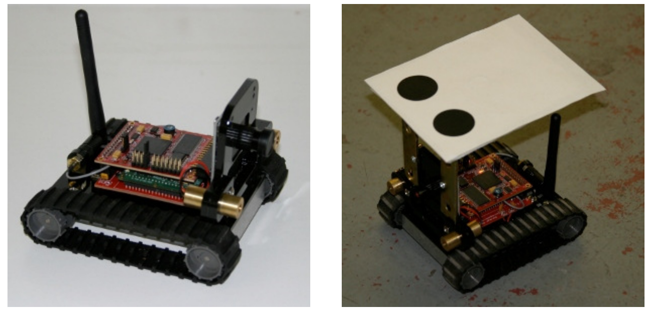
The work we report here involved a single human operator interacting with a team of three robots. In this section we describe the physical setup of the environment in which the experiments were conducted and the way the experiments themselves were conducted.

Physical setup

Our experimental testbed models the interior of a building, with a large space including six rooms and a "hallway", which the robots explore, as described below. The physical testbed is shown in Figure 1. The full space is approximately 20 ft square.

The robots that we used for these experiments are Surveyor SRV-1 Blackfins. The Blackfin[1] is a small tracked platform equipped with a webcam and 802.11 wireless. The Blackfin is pictured in Figure 2(a). Localization is provided by a network of overhead cameras (Sklar et al. 2011), and to help these cameras identify the identical robots, each robot is provided with a unique "hat". A Blackfin wearing a hat is shown in Figure 2(b). (The hats each carry a letter from the Braille alphabet, chosen so that each letter used has no rotational symmetry so that the hat provides orientation as well as position.)

[1]http://www.surveyor.com/SRV_info.html



(a) unmodified (b) with hat

Figure 2: The Surveyor SRV-1 Blackfin.

Because the Blackfin has limited on-board processing, the controller for these robots runs off-board, communicating with the robot over 802.11. Naturally this results in some lag. The controller for all the robots on the team, plus software to allocate tasks to robots (described in (Sklar et al. 2012)), and the software that extracts robot positions from the overhead cameras, runs on a group of machines that make up the "control station" for the experiments, located next to the arena. The control station is pictured in Figure 3.

Motivation

As explained above, exploration of the physical space is a key component of the tasks that we are interested in having our human/robot team perform. As a result, we have been running experiments in which robots are allocated particular "interest points" that they have to move to. As described in (Sklar et al. 2012), there is a central component that allocates these points to the robots on a team. The robot controllers then plot a path that covers all the points that their robots have been allocated, with no knowledge of what other robots are planning to do, and the robots then simulatneously manoeuvre to those points.

Figure 3: The control station.

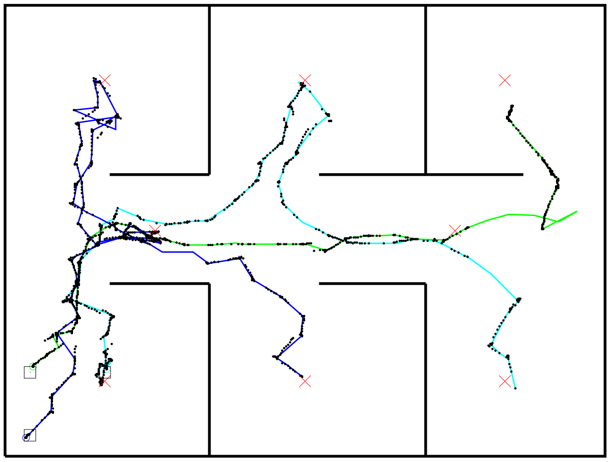


Figure 4: Three robots exploring the arena.

Given that the robots are in a restricted space, that in some experimental cofigurations the robots all start in the same part of the space (modelling situations in which the robots have all entered the space from the same point), and that robots have no knowledge of what other robots are planning to do, the robots naturally get in each other's way. This can be seen in Figure 4 which shows the motion of three robots in one experimental run. Because of this interference, the robot controller is programmed to prevent collisions, and it does this in a very conservative way — it halts one of a pair of robots that get too close to one another. (The second robot then replans its route to avoid the stationary robot, moves around it, and then the stopped robot is restarted).

Since the mechanism for preventing collisions is rather crude, it seems natural to ask if a human operator can teach the robot to do a better job of preventing collisions.

Experimental setup

The robot team was setup in the same configuration as in Figure 4. That is three robots were placed in one room in the arena (the same room as in Figure 4) and the team was allocated 5 interest points. Task allocation was carried out as described in (Sklar et al. 2012), and the robots then started to follow the paths that they had planned. The conservative collision avoidance mechanism described above was disabled.

Instead, collision avoidance was in the hands of the human teacher. This individual sat at an "operator station" that was physically remote from the control station and the arena. Indeed, for the experiments analyzed here the operator station was in a different room (a separate room in the lab complex). The teacher could not see the arena or the control station, and could not hear anything from the arena either. The only information that the teacher had about the robots was that depicted on a user interface such as that in Figure 5. This displays the current positions of the robots on a plan view of the arena along with the next point of interest that the robot is attempting to visit. The robot positions are those derived by the overhead cameras and hence are subject to error and to lag.

The teacher was given two commands that could be sent to the robots from the keyboard of their computer. The robots could be told to wait, and the robots could be told to resume movement. Each command could either be sent simulaneously to all robots, or it could be sent to a single robot. To send a command to a single robot, the teacher had to first select the robot by clicking on the icon for that robot.

Each experiment consisted of a single run of the system. The robots were positioned, points of interest were allocated, and the robots then manoeuvered to those points while being monitored for collisions by the human teacher. In each run, the teacher would make the robots wait when she thought that to be necessary to avoid a collision and then estart them when the danger of collision was paost. The run concluded either when the robots all reached their final point of interest or when robots were involved in a collision (either a collision between two robots, or a collision between a robot and a wall). These collisions were detected by operators at the control station who were also responsible for setting up the robots and running the task allocation mechanism.

Experiments and Results

The results reported here were obtained from five human teachers, each of whom was responsible for five runs. All the teachers had previously participated in similar experiments, so did not need any training runs. In order to make sure we obtained data that was relevant to learning to avoid collisions between robots, we did not use runs that ended with a robot colliding with a wall. That is, when such runs occurred, we started over, and did not count the run as one of the five for each human teacher. (As a result, some teachers were involved in more than five runs, but each only contributed five runs to the results analyzed here, those runs that ended either with the robots successfully completing their task or with a collision between robots.)

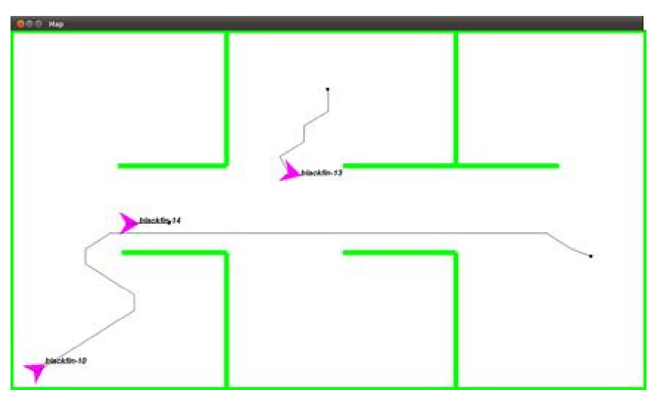


Figure 5: The user interface presented to the human teacher.

Summary

Acknowledgments

This work was supported by the National Science Foundation under #IIS-11-17000 and #CNS-0851901.

References

Blair, A. D., and Sklar, E. I. 1999. Exploring evolutionary learning in a simulated hockey environment. In *Proceedings of the 1999 Congress on Evolutionary Computation*, 197–203.

Blair, A. D.; Sklar, E. I.; and Funes, P. 1998. Co-evolution, determinism and robustness. In *Simulated Evolution and Learning*, volume 1585 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag. 389–396.

Cypher, A. 1991. Eager: Programming repetitive tasks by example. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*.

Funes, P.; Sklar, E. I.; Juillée, H.; and Pollack, J. B. 1988. Animal-animat coevolution: Using the animal population as fitness function. In *Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior*, 525–533. MIT Press.

Habib, M. K. 2007. Humanitarian Demining: Reality and the Challenge of Technology. *Jrnl of Adv Robotic Sys* 4(2).

Jacoff, A.; Messina, E.; and Evans, J. 2000. A standard test course for urban search and rescue robots. In *PerMIS Proc*.

Maes, P. 1994. Agents that reduce work and information overload. *Communications of the ACM* 37(7):31–40.

Murphy, R. R.; Casper, J.; and Micire, M. 2001. Potential tasks and research issues for mobile robots in robocup rescue. In *Robot Soccer World Cup IV, LNAI 2019*. Springer.

Santana, P. F.; Barata, J.; and Correia, L. 2007. Sustainable Robots for Humanitarian Demining. *Int Jrnl of Advanced Robotic Sys* 4(2):207–218.

Sklar, E. I., and Icke, I. 2009. Using simulation to evaluate data-driven agents. In *Multi-agent Based Simulation IX*, volume 5269 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag.

Sklar, E. I.; Salvit, J.; Camacho, C.; Liu, W.; and Andrewlevich, V. 2007. An agent-based methodology for analyzing and visualizing educational assessment data. In *Proceeding of the Sixth International Conference on Autonomous Agents and Multiagent Systems*.

Sklar, E. I.; Ozgelen, A. T.; Muñoz, J. P.; Gonzalez, J.; Manashirov, M.; Epstein, S. L.; and Parsons, S. 2011. Designing the HRTeam framework: Lessons learned from a rough-'n-ready human/multi-robot team. In *Proceedings of the Workshop on Autonomous Robots and Multirobot Systems*.

Sklar, E. I.; Ozgelen, A. T.; Schneider, E.; Costantino, M.; Muñoz, J. P.; Epstein, S. L.; and Parsons, S. 2012. On transfer from multiagent to multi-robot systems. In *Proceedings of the Workshop on Autonomous Robots and Multirobot Systems*.

Sklar, E. I.; Blair, A. D.; and Pollack, J. B. 2001. Training intelligent agents using human data collected on the internet. In *Agent Engineering*. Singapore: World Scientific. 201–226.

Sklar, E. I. 2000. *CEL: A Framework for Enabling an Internet Learning Community*. Ph.D. Dissertation, Department of Computer Science, Brandeis University.

Teitelman, W. 1979. A display oriented programmer's assistant. *International Journal of Man-Machine Studies* 11:157–187.