Comparing Human and Automated Agents in a Coordinated Navigation Domain

University of Rochester Technical Report #989

Daniel Scarafoni[1], Mitchell Gordon[1], Walter S. Lasecki[1], and Jeffrey P. Bigham[2]

Department of Computer Science[3]
University of Rochester
dscarafo@u.rochester.edu, m.gordon@rochester.edu
wlasecki@cs.rochester.edu

Human-Computer Interaction Institute[2]
Carnegie Mellon University
jbigham@cmu.edu

Abstract

From vacuum cleaners to medical assistant robots, autonomous agents are becoming increasingly common in real-world spacial settings where both people and machines are present. In this paper, we take initial steps towards understanding how these systems can coordinate with one another, even in situations where agents don't know each other's internal model of the world. Our work explores how teams of human agents and teams of AI agents function in task settings where no communication is possible. We show that there are no significant differences in the optimality of the joint paths taken by human and automated teams; this suggests that combining the two into hybrid teams would be possible without the need to compensate for shortcomings in either group. We then outline how structures for coordinating hybrid teams of humans and machines might best be able to utilize the underlying strengths of each, in order to achieve better performance.

Introduction

While applications are limited by the current capabilities of artificial intelligence (AI), recent work has shown that crowdsourcing can be used to create deployable intelligent systems that leverage human intelligence on-demand (Lasecki et al. 2011). There is significant prior literature exploring coordination of automated agents in these settings, but no work so far has focused on the performance and coordination of crowd-powered systems in multi-agent coordination settings (Barrett and Stone 2012; Barrett et al. 2013; Barrett, Stone, and Kraus 2011; Stone et al. 2010). In this paper, we present initial work in this area that compares (both qualitatively and quantitatively) the innate performance of ad-hoc human teams in spacial coordination tasks to that of teams of AI agents.

To test team coordination, we also present the ShapeTester domain. A variant of the classic pursuit domain in multi-agent systems (MAS), the ShapeTester domain asks workers to create a formation in a grid world (Undeger and Polat 2010). This allows us to focus on the coordination ability of teams in static tasks, not dynamic tasks, such as pursuit domain. Our results show that our automated agents do not perform significantly better than crowd workers recruited from Mechanical Turk in terms of move optimality. They do outperform human agents, however, in time needed for task completion. We also find that much of the time difference can be accounted for by player misunderstandings and lack of understanding of the game environment, and that the cooperation of human players depends on the incentive system used in the task. We also determined that the AI agents, despite being provably optimal in the solo case, were significantly less optimal than predicted because of the constant readjustments to their paths needed to avoid collisions with one another.

Our findings suggest that humans can coordinate in these domains even without explicit support or communication channels. Finally, we conclude with a discussion of future research on this topic that aims to explore the ability of humans and machine to jointly form *hybrid* teams that can perform as all or even better than teams with homogeneous sets of workers by leveraging the strengths of each group.

Related Work

Our work builds on research in artificial intelligence (multi-agent systems and agent models), computer supported cooperative work (remote human teams), and crowdsourcing (interactive crowd-powered systems and crowd agents).

Ad-Hoc Teams of Automated Agents

Prior work in Multi-Agent Systems (MAS) has explored development of machine-learning techniques (such as bayesian inference and applications of game theory) in an attempt to develop agents which can perform well in pursuit domain, autonomous navigation, and other domains (Lasecki et al. 2013). Real-world applications for multi-agent systems are readily apparent, and include improved efficiency from coordination in military drone activity and coordinated disaster response by automated robots (Barrett and Stone 2012; Barrett et al. 2013; Barrett, Stone, and Kraus 2011). However, no studies have been done on the role of human agents in ad-hoc teams (Stone et al. 2010).

Models Barrett et al proposed the multi-agent challenge of creating an agent which is able to collaborate with other agents in ad-hoc environments in an extensible and a robust manner (Stone et al. 2010). Since then, models used for this set of problems have been diverse, though are usually presented as concrete game problems (Barrett et al. 2013; Benda, Jagannathan, and Dodhiawala 1986). Nasroullahi et al used a continuous domain over which several robotic agents observed points of interest; Agmon et al worked without a concrete implementation, choosing instead to evaluate the problem in terms of abstract state-space. Because we wanted to involve real people in our investigation, a more game-like and accessible experimental domain was necessary. As such, many of the potential domains were poorly suited for our experiments.

Experimental Domains Since multi-agent environments are often expensive to setup and test in the real-world, we introduce a simulated domain that embodies many of the problems we're interested in solving (Stone et al. 2010). Prior work has explored the *pursuit domain* where one or more *predator* agents attempt to catch a number of *prey* agents (Benda, Jagannathan, and Dodhiawala 1986). The specifics of the setup can differ considerably. Undeger et al. used a single prey with multiple predator agents pursuing it, and used strategies such as blocking escape directions (BES), in which predator agents try to find an open square which ”cuts off” prey movement. Techniques such as this are used to evaluate the efficiency of the predators (Undeger and Polat 2010). Barret and Stone utilize a number of predators (four) to capture one prey, though they evaluate agents on an individual level (Barrett, Stone, and Kraus 2011).

In our setup, we introduce the *ShapeTester* problem, which is closely related to the pursuit domain. In this domain, we utilize a toroidal world; if an agent went over the edge on one side, they would reappear on the other. One type of agent, predator, was implemented. The dynamic aspect was removed to better control the problem for observation. Further, our experimentation focused on implementing the ”surround” method of capture, and as such we were able to simplify and generalize the domain (Stone et al. 2010; Undeger and Polat 2010). Instead of testing the ability of agents to surround a prey, we evaluated the ability of these agents to form arbitrary shapes in general. Our experimentation also differed in that the efficiency of various pathfinding algorithms were not tested (Barrett, Stone, and Kraus 2011; Undeger and Polat 2010).

Human Teams

The ShapeTester problem makes use of remote operation and telepresence. Workers view a live feed of a game board and remotely control the movement of their assigned piece on it. Remote control is a frequent subject in past studies. For example, ShareCam introduced the idea of combining the input of multiple remote users in order to control the movement of a camera (Song and Goldberg 2003). Our work differs from this approach in that we assign each worker their own individual item to control. Though it does test the ability of many users to cooperate without communication, it does not treat each user as an individual agent, as our domain does. Closer to our system is one introduced by Osentoksi, et al., who present a tool that shows a live video feed of a robot to a user, and allows the user to remotely control the movement of the robot (Osentoski et al. 2010). This is somewhat like a real-world representation of our all-digital system for ShapeTester.

Crowdsourcing and Crowd Agents

Crowdsourcing uses humans in the form of online workers to accomplish computational tasks (von Ahn 2005). Our ShapeTester problem requires the use of a real-time, synchronous continuous crowd (Lasecki and Bigham 2013), in which we keep workers engaged for longer periods of time so that they can successfully complete the ShapeTester game. Other applications of the continuous crowd include Legion:AR, which asks workers to label activities in videos in real time(Lasecki et al. 2013).

In order to conduct our real-time experiment, we need many online crowd workers to begin a task simultaneously. To accomplish this, we use the *retainer model* (Bernstein et al. 2011). This model allows us to produce a pool of online workers by recruiting them on Amazon Mechanical Turk and then alerting them when our task was available to begin. One challenge of the retainer model is ensuring that a sufficient number of workers are in our retainer pool. Prior work has addressed this challenge and explored the rates at which workers arrive and leave a retainer. For example, Bernstein et al. describe the relationships between task and worker arrival rates, size of the retainer pool, cost, and expected wait time for workers (Bernstein et al. 2012).

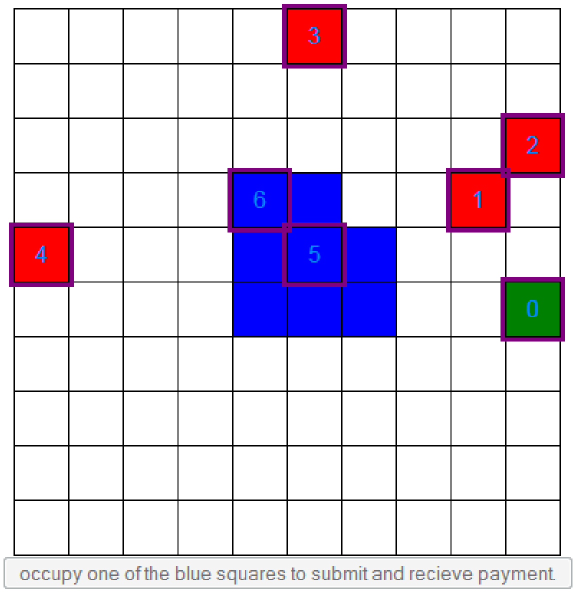
In addition to pure crowdsourcing, our work also explores the idea of merging AI and human workers. Prior work in this area has discussed how the use of autonomous agents could reduce cost and raise quality of results (Weld 2010).

The Shape Tester Domain

Our domain is designed to test the efficacy of human workers vs artificial intelligence (AI) agents in an ad-hoc, low communication setting. Much work has been done developing optimal AI agents for a number of domains (Genter, Agmon, and Stone 2011). However, workers can be quickly recruited for team-tasks. Our study can gather one experiment's worth of workers from Mechanical Turk in less than one minute, and pay them each 25 cents for the task. This, of course, begs the question of the trade-off between workers and artificial agents. Our study is the first to attempt to answer this question.

Players are represented as colored squares on a game board (Figure 1). The overall objective of the ShapeTester task is for all agents to occupy specified ”critical spaces.” These spaces are arranged to form a given shape, thus simulating the ”capture” condition of previous experiments (Barrett and Stone 2012). The *critical spaces* (the spaces that need to be occupied in order to complete the task) are colored blue.

Our testing domain was a 10 by 10 grid. The game was torroidal: when a player moves off of one side of the map,



occupy one of the blue squares to submit and recieve payment

Figure 1: An example of the ShapeTester board mid-game from the perspective of player 0. Other players appear as red squares, goal spots are blue squares. The players must navigate to the blue squares to form the shape and complete the task.

they reappear on the other. Players saw themselves as green, and saw other players as red. A player's ID is written on their square. When a player occupies a critical space, his or her piece turns blue (though an outline was used to indicate that the space was occupied). An image of our setup can be seen in Figure 1. Our setup utilizes two separate sources for players: workers recruited from Amazon's Mechanical Turk, and optimal artificial agents. The game terminates when all critical spaces were occupied.

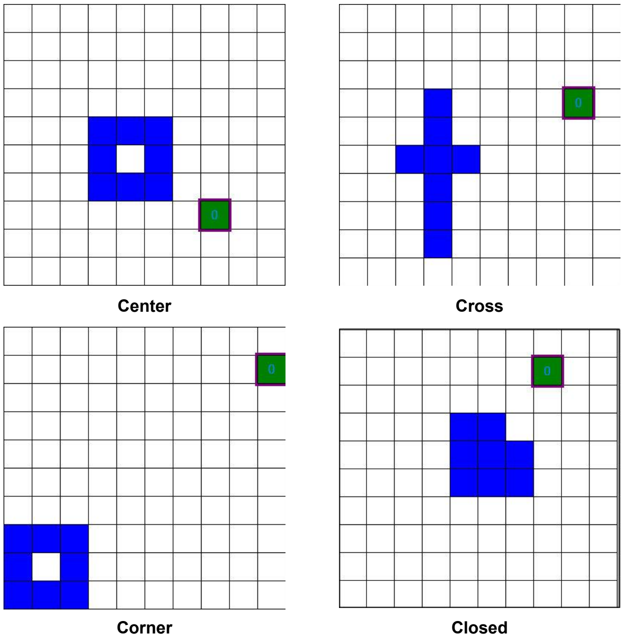
By having a static target (spaces to occupy) instead of a moving one, we are better able to control the environment, and thus measure the optimality of the players more efficiently. This setup emphasized the core ”capture” aspect of the domain. Our setup utilizes eight players (as opposed to the conventional four) to add an increased emphasis on teamwork, and to allow more complex shapes to be formed.

Experimental Setup

Our primary measure of our experiments was move optimality. This is defined as the minimum number of moves needed to move from the starting position of a player to their ending position divided by total moves taken (in game). To test this, the starting position, ending position, and each move made in between is recorded as the players moved. To ensure that coordination was necessary in-game, all players are placed at random starting locations in the game world at the same time at the beginning of the experiment.

Four shapes are tested in this study (Figure 2):

- The first, *center*, is a hollow rectangle in the center of the map. It most closely resembles the 4-predator trap shape discussed earlier, and is meant to test this scenario.



Center

Cross

Corner

Closed

Figure 2: The four shapes used in the ShapeTester game. The shapes are meant to be diverse and pose different obstacles. Closed and cross allow player configurations in which certain critical spots would become inaccessible if players were not coordinated.

- *corner* is a rectangle (same shape as center) in the corner of the map. Its location tests human players ability to reason on a toroidal world. Given this configuration, it is very easy for paths to critical spots to be blocked by other players, meaning players have to make use of the game's ”wrapping” in order to finish the task.
- The next shape, *closed*, is similar to center except that the inner square of the shape was a critical spot as well. This is meant to experiment with the players' ability to coordinate in a scenario where certain critical spots can be inaccessible if players do not plan accordingly.
- The final shape, *cross* is a cross in the center of the map. Like *closed*, this shape is meant to test the players' abilities when certain critical spots can become inaccessible. This shape is also very long, and the ”wall” created by players in the critical spots helps evaluate other players' ability to navigate around obstacles.

Results

Overall, we find that human and AI agents are able to find paths to their end-goals with similar efficiency. AI teams, however, are much faster than human ones.

Task Ability

AI agents are able to finish the task at hand in all but two tests. By this we mean experiments with empty critical spots which could not be reached because other players were blocking the path. Deadlocks are only possible in the closed

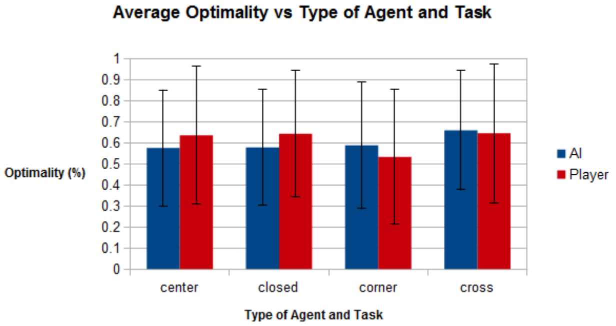


Figure 3: Human and AI agents have similar move optimality across all tests.

and cross experiments, and both of the deadlocks mentioned happened in the latter test.

All but four human tests finished. If a human agent did not move for more than two minutes, they were assumed to have abandoned the task, and an AI agent was sent in to finish the game so that the agents could be payed. We observe that many humans contained *slow-learning* agents. These individuals typically took much longer (more than 30 seconds) to finish the task than other players. 35 out of 40 tests had at least one of these players.

Comparing Team Performance

Our results show that human and AI agents have comparable move optimality, and that humans take significantly longer time to finish the task than AI agents.

Move Optimality

Optimality analysis indicates that there is not significant difference between human and AI agents in terms of the number of moves needed to finish the task. Two-tailed t-tests all return values $P < 0.05$. This holds true across all four tests.

The optimality of the human agents is higher than the AI in the center and closed tests, though not significantly so. The AI optimality is higher for the remaining two tests. This difference is likely caused simply by statistical anomaly. Optimality results all report values of $F > 0.05$. Results show no change in optimality variance between human and AI agents. Figure 3 shows the results.

Time to Completion

Time analysis provides a different perspective. An analysis of the average time needed for each task shows the clearly superior performance of AI agents over human. The human agents show considerably more variation in the time needed to complete their tasks. Even so, they are, on average, several magnitudes of ten slower than their AI counterparts. A two-tailed t-test results in a value of $P < 0.0001$ for all tests (when comparing human and AI times).

F-tests return similar results; AI agents show much less variance than human agents. This is consistent with previous observations: most human players are able to finish the task in a matter of seconds, but a small number take a much longer period of time.

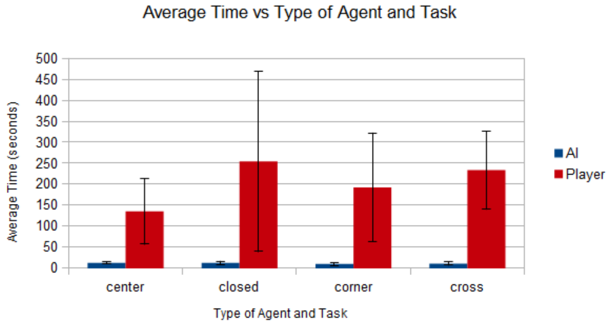


Figure 4: Human agents showed more variation in the time needed to complete the task, but still took much longer than AI.

The human agents need the longest time on the closed and cross maps to complete the task. This is likely because these two maps allow players to block one another from reaching critical spots. The average time for the corner map is also higher than the center map. We attribute this to the confusion surrounding the wrap-around mechanic, which is essential to solving this map. Figure 4 illustrates these results.

Qualitative Differences

Human beings' behavior patterns change dramatically depending on economic incentives. For example, the task originally payed players when they occupied a spot. Players would regularly rush towards the nearest spot and leave; this cut off players from other critical areas and, too often, made the game impossible to finish. After changing the game such that players would only be payed when the task was completed, players left less frequently, and even helped other players (moving out of the way to let other players take a closer spot). There are even instances of players moving their pieces to guide other struggling players, trying to coax them to occupy a spot. We attribute this to purely economics reasons: the players began acting this way when they began having personal stakes in their partners' performance.

Further, human players struggle significantly with the toroidal aspect of the map. AI agents are able to easily take the shortest path, regardless of how many times they needed to go off one edge of the map. Human players, however, are often perplexed by this property. Many refuse to utilize the wrap-around feature, even after witnessing several players do so. We attribute this to a combination of a lack of instruction-reading on the part of the players, and a naive assumption that the game board behaved like a chess board (which is superficially resembles).

Another note is the difference between normal and *slow-learning* players. As noted above, the former are able to accomplish the task quickly, but the latter take much longer. Normal players would finish in under 10 seconds, but *slow-learning* players would often take several minutes to finish.

Discussion

With regard to the original question our study posed, "can agents work together in ad-hoc situations with no communication?", our results show that both human and AI agents are able to accomplish this task. In the AI case, they are even able to do so without any need for teamwork algorithms. We are also able to gage the efficacy of these two types of agents in these situations. Our study shows that their move optimality is comparable, but that the AI agents are still able to accomplish the task at hand much faster than humans.

Further, our results indicate that AI agents can be combined with human agents to finish tasks. The main limitation for hybrid teams is the significant difference in average time needed between human and AI agents to finish the task. The extra time needed by human teams, however, is mostly due to the *slow-learning* players. If these players are removed, the human teams would perform similarly to AI teams. Even so, it is not known if integrating faster AI agents with human teams would disrupt move optimality, and it may be possible to integrate the two as is.

Future task-designers will likely have to compensate for these observations. Improvements to the user interface and instructions could lessen the learning curve for the task, and allow human teams to finish quicker. The volume of *slow-learning* players could also be reduced with a ”membership” bonus: an incentive or reward for experienced players to return to play the game.

Further limitations exist with testing methods. The ShapeTester domain is a narrow domain, and the problem boils down to pathfinding. The optimal algorithms for this are, in the solo case, only O(nlog(n)). Given a much more complex (and possibly NP-complete) task, it is unknown if the AI agents would still be able to outperform humans in terms of time, or if they would be able to keep pace in terms of move optimality. Further, it was noted that human performance was negatively affected by a lack of knowledge and understanding of the game board, and that human player behavior was affected by the incentive system used. Thus, it could be that human players exhibit significantly different behavior (and therefore different results) if the user interface is changed.

Future Work

We plan to extend this work to include tests of *hybrid* teams, which include a combination of human and AI agents. In addition to the initially even performance of humans versus AI in our results, prior work has proposed a number of ways to use AI to improve the performance of crowd workers (Dai, Weld, and others 2011; Lasecki et al. 2012a; Weld 2010; Naim et al. 2013; Talamadupula et al. 2013). Continuing work will thus explore not only how to coordinate hybrid teams, but how to use the knowledge of one group to fill in for the shortcoming of the other.

The coordination task can also potentially be enhanced by finding sub-structures in teams. By using existing methods of finding similar graph subcomponents, interaction paradigms can be found and used to better anticipate the actions of other agents, even in settings where no explicit communication is allowed (Koutra et al. 2011). Since it has been shown that crowd workers also remember task specifics despite completing large numbers of tasks, it might also be possible to even designate coordination patterns between hybrid sets of agents (Lasecki et al. 2012b).

Conclusion

Our experiments show a disconnect between human and AI agents with the respect to the pursuit domain performance. Human agents are shown to have similar move optimality to AI, which indicates that human agents can serve as substitute for AI agents in situations where move number is the prime concern.

However, the human agents take much longer than AI agents. This means that any situations in which rapid solvency of the task at hand is paramount, the task is better suited to AI agents, rather than humans. This may not hold true if the *slow-learning* players are removed from the game. It should also be noted that the AI agents were artificially limited in the their speed (moves per second), and in fact could be even faster. In tasks where time completion is most important, the AI agents are clearly the superior means of completing the task.

Further testing is needed to find the exact situations and areas where human agents are best used; there were several variables that were not controlled in our experiments. For example, the large difference in time was mostly because of the time the human players needed to learn the basics of the game. We hope to continue experimentation with human and AI teams to find the most effective combination of human and machine intelligence for effective task completion.

References

Barrett, S., and Stone, P. 2012. An analysis framework for ad hoc teamwork tasks. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 357–364. International Foundation for Autonomous Agents and Multiagent Systems.

Barrett, S.; Stone, P.; Kraus, S.; and Rosenfeld, A. 2013. Teamwork with limited knowledge of teammates.

Barrett, S.; Stone, P.; and Kraus, S. 2011. Empirical evaluation of ad hoc teamwork in the pursuit domain. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, 567–574. International Foundation for Autonomous Agents and Multiagent Systems.

Benda, M.; Jagannathan, V.; and Dodhiawala, R. 1986. On optimal cooperation of knowledge sources - an empirical investigation. Technical Report BCS–G2010–28, Boeing Advanced Technology Center, Boeing Computing Services, Seattle, WA, USA.

Bernstein, M. S.; Brandt, J. R.; Miller, R. C.; and Karger, D. R. 2011. Crowds in two seconds: Enabling realtime crowd-powered interfaces. In *Proceedings of the 24th annual Association for Computing Machinery symposium on User interface software and technology*, User Interface Software and Technology '11, 33–42. New York, NY, USA: Association for Computing Machinery.

Bernstein, M. S.; Karger, D. R.; Miller, R. C.; and Brandt, J. R. 2012. Analytic methods for optimizing realtime crowdsourcing. In *Proceedings of Collective Intelligence*, CI 2012.

Dai, P.; Weld, D. S.; et al. 2011. Artificial intelligence for artificial artificial intelligence. In *Twenty-Fifth Association* *for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*.

Genter, K. L.; Agmon, N.; and Stone, P. 2011. Role-based ad hoc teamwork. In *Plan, Activity, and Intent Recognition*.

Koutra, D.; Parikh, A.; Ramdas, A.; and Xiang, J. 2011. Algorithms for graph similarity and subgraph matching.

Lasecki, W. S., and Bigham, J. P. 2013. *Handbook of Human Computation*. Springer. chapter Interactive Crowds: Real-Time Crowdsourcing and Crowd Agents.

Lasecki, W.; Murray, K.; White, S.; Miller, R. C.; and Bigham, J. P. 2011. Real-time crowd control of existing interfaces. In *Proceedings of the 24th annual Association for Computing Machinery symposium on User interface software and technology*, User Interface Software and Technology '11, 23–32.

Lasecki, W. S.; Bigham, J. P.; Allen, J. F.; and Ferguson, G. 2012a. Real-time collaborative planning with the crowd. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Lasecki, W.; White, S.; Murray, K.; and Bigham, J. 2012b. Crowd memory: Learning in the collective. In *Proceedings of Collective Intelligence 2012*, Computational Intelligence 2012.

Lasecki, W. S.; Song, Y. C.; Kautz, H.; and Bigham, J. P. 2013. Real-time crowd labeling for deployable activity recognition. In *Proceedings of the International ACM Conference on Computer Supported Cooperative Work and Social Computing*, Computer Supported Cooperative Work 2013.

Naim, I.; Gildea, D.; Lasecki, W.; and Bigham, J. P. 2013. Text alignment for real-time crowd captioning. *Proceedings of North American Association of Computational Linguistics Human Language Technologies* 201–210.

Osentoski, S.; Crick, C.; Jay, G.; and Jenkins, O. C. 2010. Crowdsourcing for closed loop control.

Song, D., and Goldberg, K. 2003. Sharecam part i: Interface, system architecture, and implementation of a collaboratively controlled robotic webcam.

Stone, P.; Kaminka, G. A.; Kraus, S.; Rosenschein, J. S.; et al. 2010. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *Association for the Advancement of Artificial Intelligence*.

Talamadupula, K.; Kambhampati, S.; Hu, Y.; Nguyen, T. A.; and Zhuo, H. H. 2013. Herding the crowd: Automated planning for crowdsourced planning. In *First AAAI Conference on Human Computation and Crowdsourcing*.

Undeger, C., and Polat, F. 2010. Multi-agent real-time pursuit. *Autonomous Agents and Multi-Agent Systems* 21(1):69–107.

von Ahn, L. 2005. *Human Computation*. Ph.D. Dissertation, Carnegie Mellon University, Pittsburgh, PA.

Weld, P. D. M. D. S. 2010. Decision-theoretic control of crowd-sourced workflows. In *Twenty-Fourth Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*.