Asynchronous Control with ATR for Large Robot Teams

Nathan Brooks, Paul Scerri, Katia Sycara
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15260 U.S.A.
nbb@andrew.cmu.edu, pscerri@cs.cmu.edu,
katia@cs.cmu.edu

Huadong Wang, Shih-Yi Chien, Michael Lewis
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA 15260 U.S.A.
huw16@pitt.edu, shc56@pitt.edu,
ml@sis.pitt.edu

In this paper, we discuss and investigate the advantages of an asynchronous display, called "image queue", tested for an urban search and rescue foraging task. The image queue approach mines video data to present the operator with a relevant and comprehensive view of the environment by selecting a small number of images that together cover large portions of the area searched. This asynchronous approach allows operators to search through a large amount of data gathered by autonomous robot teams, and allows comprehensive and scalable displays to obtain a network-centric perspective for unmanned ground vehicles (UGVs). In the reported experiment automatic target recognition (ATR) was used to augment utilities based on visual coverage in selecting imagery for presentation to the operator. In the *cued* condition a box was drawn in the region in which a possible target was detected. In the *no-cue* condition no box was drawn although the target detection probability continued to play a role in the selection of imagery. We found that operators using the image queue displays missed fewer victims and relied on teleoperation less often than those using streaming video. Image queue users in the *no-cue* condition did better in avoiding false alarms and reported lower workload than those in the *cued* condition.

INTRODUCTION

The task of interacting with multi-robot systems (MrS), especially with large robot teams, presents unique challenges for the user interface designer. These challenges are very different from those arising in interactions with a single or limited number of robots. Traditional graphical user interfaces and infrastructures have difficulties in interacting with a large MrS. The core issue is one of scale: in a system of *n* robots, any operator task that has to be done for one robot must also be done for the remaining *n*–1 robots (McLurkin et al., 2006). The interface for a large robot team needs to simultaneously provide for command and coordination of distributed action while centralizing and integrating the display of data.

Many different applications, such as interplanetary construction, search and rescue in dangerous environments, or cooperating unmanned aerial vehicles, have been proposed for MrS. Controlling these robot teams has been a primary concern of many human-robot interaction (HRI) researchers. These efforts have included the use of both the theoretical and applied development of the Neglect Tolerance (Crandall et al., 2006) and Fan-Out models (Olsen & Wood, 2006) to characterize the control of independently operating robots; predefined rules to coordinate cooperating robots, as in Playbook™ (Miller and Parasuraman, 2007) and Machinetta (Scerri et al., 2005); and techniques for influencing teams obeying biologically inspired control laws (Kira & Potter, 2010). While our efforts to increase the span of control over unmanned vehicle (UV) teams appear to be making progress, the asymmetry is growing between what we can command and what we can comprehend.

Automation can reduce excessive demands for human input, but throttling the information being collected and returned is fraught with danger. A human is frequently included in the loop of a MrS to monitor and interpret the video that is being gathered by UVs. This can be a difficult task for even a single camera (Cook et al., 2006) and begins to exceed operator capability before reaching ten cameras (Lewis et al., 2010). With the increasing autonomy of robot teams and plans for biologically-inspired robot "swarms" of much greater number, the problem of absorbing and benefiting from their output seems even more important than learning how to command them.

Foraging tasks, when carried out with a large robot team, usually require a more detailed exploration than simply moving each robot to different locations in the environment. Acquiring a specific viewpoint of targets of interest (e.g. finding victims in a disaster scenario) is of greater concern, and increasing the explored area is merely a means to achieve this end. While a great deal of progress has been made in the area of autonomous exploration, the identification of targets is still typically done by human operators who ensure that the area covered by robots has in fact been thoroughly searched for the desired targets. Without the means to combine the data gathered by all of the robots, the human operator is required to synchronously monitor their output, such as by using a video feed for each robot. This requirement and load on the human operator may directly conflict with other tasks, especially navigation which requires the camera to be pointed in the direction of travel in order to detect and avoid objects. The need to switch attention among robots will further increase the likelihood that a view containing a target will be missed. Earlier studies (Pepper et al., 2007) confirmed that the search performance of these tasks is directly related to the frequency with which the operator shifts attention between robots, and is possibly due to targets missed in the video stream while servicing other robots.

The problem addressed in this paper is the design of an asynchronous, scalable, and comprehensive display, without requiring a 3D reconstruction, to enable operators to detect relevant targets in environments that are being explored by large teams of unmanned ground vehicles (UGVs). We will present one particular design for such a display and test it in the context of Urban Search and Rescue (USAR) by using large robot teams that have some degree of autonomy and are supervised by a single operator.

ASYNCHRONOUS IMAGERY

An asynchronous display method can alleviate the concurrent load put on the human operator and can disentangle the dependency of tasks that require direct attention to multiple video feeds. Furthermore, it is possible to avoid attentive sampling among cameras by integrating multiple data streams into a comprehensive display. In turn, this allows the addition of new data streams without increasing the complexity of the display itself. An earlier approach to asynchronous display for USAR was explored in (Velagapudi et al. 2008). The method, motivated by asynchronous control techniques previously used in extraterrestrial NASA applications relied on substituting a series of static panoramas taken at designated locations for continuous video. The operator then searched through the panoramic images to determine the location of targets viewable from each of the selected locations. In a four robot experiment comparing panoramas with streaming video there was no difference in the number of victims found or area explored. A further experiment (Velagapudi et al., 2009) scaled the team size to eight and twelve robots on the premise that advantages for self paced search of imagery might emerge with increasing numbers of video feeds to monitor in the synchronous control condition. Again, no differences were found. However, this approach did not utilize all the available data from the video feeds that robots gather, so a huge amount of potentially useful information in the panorama condition was discarded. Furthermore, the operator must give the robots additional instructions on where to sample future panoramas.

In contrast to previous work, the present approach allows the use of autonomous exploration. We present an asynchronous display that mines all of the robot video feeds for relevant imagery, which is then given to the operator for analysis. We call this type of asynchronous display “image queue” and compare it to the traditional synchronous method of streaming live video from each robot, which we refer to as “streaming video”. In the next section, we describe our test bed, along with a detailed description of the image queue and a comparison with streaming video.

The goal of the image queue interface is to use the advantages of an asynchronous display and to maximize the amount of time human operators can spend on the tasks that humans perform better than robots. For USAR, this is currently the case for tasks like victim identification and navigating robots out of dangerous areas in which they are stuck. As the number of robots in a system increases with improved autonomy, the demands on operators for these tasks increase as well. Hence, another requirement for the interface is to provide the potential for scaling up to larger numbers of robots and operators. The proposed image queue interface implements the idea of asynchronous monitoring via a priority queue of images that allows operators to identify victims requiring neither synchronicity nor any contextual information not directly provided by the image queue.

The image queue interface (Fig. 1) focuses on two tasks: (1) viewing imagery, and (2) localizing victims. It consists of a filmstrip viewer designed to present the operator with a filtered view of what has passed before the team’s cameras. A filtered view is beneficial, because the video taken contains a high proportion of redundant images from sequential frames and overlapping coverage by multiple robots.

ATR

The ATR (automatic target recognition) algorithm uses prior knowledge of a victim’s visual appearance, such as shirt color, pants color, or skin tone, to calculate an image's victim probability. In the *cued* condition, the probability P of a victim being present is equal to the sum of pixels in the image, correlated with victim colors, divided by a tuned threshold, which is bounded to [0.1, 1]. In addition, a target indicator (Fig. 1) was added in the *cued* condition to assist the operator in identifying the detected target (victims). If an image's victim probability was greater than 30%, a bounding box was drawn around the victims estimated location. Because color histogram-based target detection proved unrealistically accurate when used with synthetic imagery, false alarms with a rate consistent with that expected for real imagery were introduced. If an image's victim probability was less than 30%, a false positive was generated with a 20% probability and a randomly generated victim bounding box was drawn on the image.

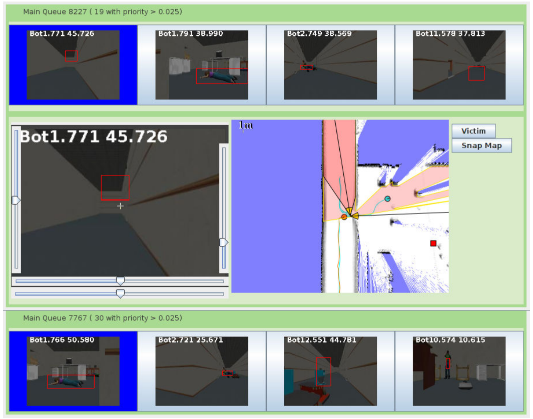

Figure 1. Image Queue GUI with Target Detection

IMAGE UTILITY

Next, the visual coverage of an image is computed by referencing the image in the map, as seen in Fig. 2. From this we compute a coverage utility score U. Images covering larger areas, excluding parts already been seen by other images, receive higher utility scores. In colloquial terms this kind of utility ranks images higher that cover large areas with minimal overlap. Fig. 2 illustrates this concept of utility with a simple example. We normalize utility to the bounded interval [0, 1].

The image is now added to the priority queue, with priority equal to P*U (victim probability* coverage utility). As the operator views images, the utility is recalculated to take into account the growing portion of the world the operator has viewed, causing the priority of the image to be recalculated and the queue to be rearranged as necessary.

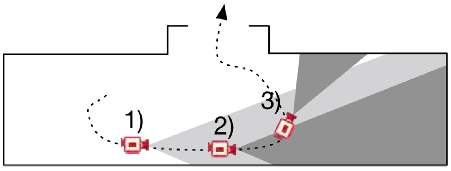

Figure 2. Determining utility for Image Queue

By aggregating imagery with the highest priority scores at regular intervals, the image queue allows the operator to peruse a relatively small number of prioritized images that show most of the new area explored by the robots that is likely to contain targets. Notice that exploration can continue while operators view the image queue, as long as robots are sufficiently autonomous (or controlled by other operators). Operators can either click or scroll through a certain number of images in the queue. Once operators work through the first set of images, the image queue marks the areas covered by these images as already seen and retrieves the next set of images with high utility. Tests of this system show that after 15 minutes of exploration, an operator can view 70% of the area covered by viewing the 10 highest utility frames, and 90% of the area covered within the first 100 frames.

METHODS

USARSim and MrCS

The experiment reported in this paper was conducted using the USARSim robotic simulation with 12 simulated Pioneer P3-AT robots performing Urban Search and Rescue (USAR) foraging tasks. USARSim is a high-fidelity simulation of USAR robots and environments that was developed as a research tool for the study of human-robot interaction (HRI) and multi-robot coordination. USARSim supports HRI by accurately rendering user interface elements (particularly camera video), accurately representing robot automation and behavior, and accurately representing the remote environment that links the operator's awareness with the robot's behaviors. USARSim also serves as the basis for the Virtual Robots Competition of the RoboCup Rescue League.

MrCS (Multi-robot Control System), a multi-robot communications and control infrastructure with accompanying user interface, developed for experiments in multi-robot control and RoboCup competition ("Robocup Rescue VR", 2010) was used in this experiment. MrCS provides facilities for starting and controlling robots in the simulation, displays multiple camera and laser output, as well as maps, and supports inter-robot communication through Machinetta, a distributed multi-agent coordination infrastructure. Fig. 3 shows the elements of the conventional GUI for the streaming video condition. The operator selects the robot to be controlled from the colored thumbnails, with live videos appearing at the top right of the screen. The current locations and paths of the robots are shown on the Map Viewer (bottom left). When under manual control, robots are tasked by assigning waypoints on a heading-up map on the Map Viewer or through the teleoperation widget (lower right).

An autonomous path planner was used in the current experiment to drive the robots, unlike the panorama study (Velagapudi et al. 2008) in which paths were manually generated by participants with specified panorama locations. As in the previous study (Chien et al. 2010), operators appeared to have little difficulty in following these algorithmically generated paths, and identified approximately the same numbers of victims (per operator) as those following human generated paths.

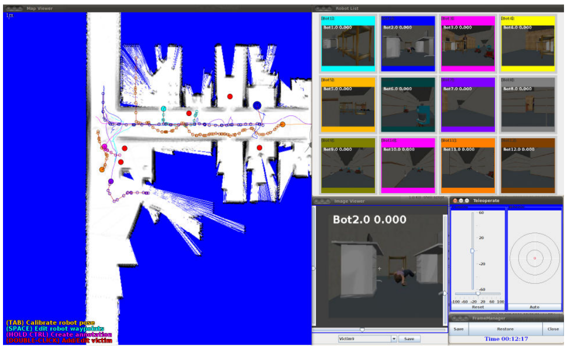
(TAB) Calibrate robot pose
(SPACE) Edit robot waypoints
(HOLD CTRL) Create annotation
(DOUBLE-CLICK) Add/Edit victim
Bot2.0 0.000
Time 00:12:17

Figure 3. GUI for the streaming video condition.

Participants and Procedure

30 paid participants approximately balanced by gender were recruited from the University of Pittsburgh community. Participants were provided with standard instructions on how to control robots via MrCS. In the following training session, participants practiced control operations for both streaming video and image queue conditions for 10 minutes each. After the training session, participants began the two 15-minute real-task sessions in which they performed the search task, controlling 12 robots in teams. Experiment followed a two condition repeated measures design comparing the streaming video with an image queue display with ATR. In addition, the image queue condition participants have been separated into two sub groups: Cued and Non-cued. The environment was 5026 m^2, a size sufficient to guarantee that no participant could complete exploration. There were 100 victims distributed in the environment. At the conclusion of each real task session, participants were asked to complete the NASA-TLX workload survey (Hart & Staveland 1998).

RESULTS

Data were analyzed using a repeated measures ANOVA to compare streaming video with the image queue conditions and a one-way between groups ANOVA to compare the cued and no-cue groups within the image queue condition. Participants were successful in searching the environment with no significant differences between conditions ($F_{1,28}$ = .181, p = .674) or groups ($F_{1,28}$ = .103, p = .751). On average, participants in the streaming video condition found 9.03

victims, while those in the image queue conditions found 8.73. The area explored by the 12 robots also showed no significant differences among displays ($F_{1,28}$ = 0.479, p = .495).

Every mark that a participant made indicating a victim was compared with ground truth to determine whether there was actually a victim at the location. A mark made further than 2 meters away from any victim or multiple marks for one victim were counted as *false positives*. Victims that were missed, but present in the video feed and not marked were counted as *false negatives*. The number of false positives showed no significant difference between the image queue conditions and streaming video ($F_{1,28}$ = .053, p = .819). A one-way ANOVA, however, found a significant advantage for the *no-cue* group over the *cued* group ($F_{1,28}$ = 4.974, p = .034) within the image queue conditions.

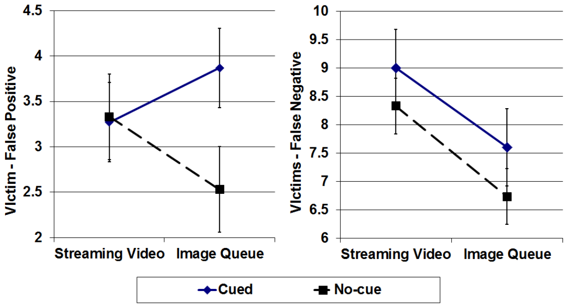

Figure 4. Marking errors for victims

The image queue did, however, show a significant improvement over the streaming video condition ($F_{1,28}$ = 7.292, p = .012) for false negatives, with the average number of missed victims dropping to 7.17 from the 8.67 missed in the streaming video condition (Fig. 4).

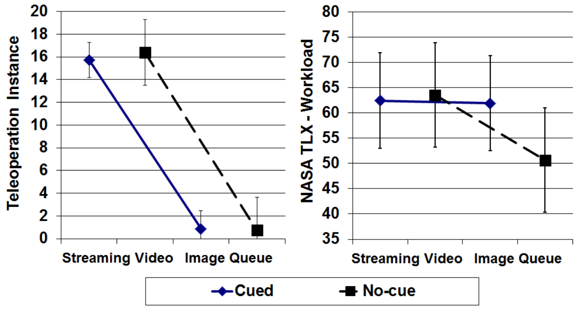

Figure 5. Teleoperation and workload

In MrCS control operators have frequently been observed (Lewis et al., 2010) to engage in teleoperation in order to regain situation awareness (SA) by finding the robot and orientation associated with a camera view. Because both image queue and streaming video users are equally likely to need to teleoperate to free stuck robots, differences in teleoperation frequency provide an indirect measure of SA. A repeated measures ANOVA shows a significant difference ($F_{1,28}$ = 176.845, p < .001) for the count of teleoperation times between the streaming video and image queue condition with participants in the streaming video condition teleoperating an average of 16.07 times while they chose to teleoperate only 0.87 times in the image queue condition.

While the full-scale NASA-TLX workload measure (Fig. 5) revealed that no advantage for either the image queue or streaming video conditions, the no-cue version of the image queue was judged significantly less taxing than the cued version ($F_{1,28}$ = 5.364, p = .028).

DISCUSSION

The purpose of this experiment was to examine the effects of the asynchronous image queue with automatic target on overall performance. It presents information to subjects asynchronously, but is ordered by a quality metric that relates to the utility of the information and the probability of finding a victim. This stands in contrast to the video stream that presents information as it becomes available. Additionally, our technical implementation of an image queue based on a ranking by priorities allows the addition of further utility criteria, such as fire and other hazards that need to be detected, depending on the particular application.

Our results show that in the image queue conditions, which allow interruption and relevant image retrieval, a reviewable, location-based image queue interface leads to similar search performance with lower operator errors and a overall lower workload. In the streaming video condition, we observed more instances of teleoperation, while participants in the image queue condition avoided teleoperating the robots and relied more heavily on autonomy. As autonomy improves, we ultimately expect to see the need for navigation reduced to situations in which the operator has to assist robots in fixing unexpected errors. Furthermore, image queue participants have no need to teleoperate a robot, in contrast to streaming video participants when they encounter a victim in the video feed. Most importantly, they do not need to stop the robot in order to precisely locate the victim. In essence, we have decoupled the navigation and error-recovery tasks from the victim-detection tasks, allowing the latter tasks to be completed entirely asynchronously without any penalties for performance in terms of the number of victims. Also, by decoupling these tasks, we reduced the number of false-negative errors that occur. The reduction in errors for the image queue condition is particularly significant, because avoiding missed targets is crucial to most foraging tasks. Thoroughness and correctness are two of the most important performance metrics, especially for USAR when lives depend on it.

When examining performance and workload and comparing the *no-cue* group with the *cued* group in the image queue condition, some unexpected but interesting results may give a hint as to the design of an appropriate interaction procedure. Originally, it was expected that the *cued* display might reduce user workload and improve overall performance. However, the analysis of victim marking errors shows that the *cued* group marked 52.9% more victims at the wrong location (false positive, Fig. 4) but did not miss more victims. A similar disadvantage in reported workload suggests that substantial cognitive resources were required for the *cued* group to separate false alarms from accurately placed boxes. An example of a final map for the *cued* group illustrates this problem (Fig. 6). When the operator viewed all images from a newly covered area or with newly detected victims, the system may continue to pull images from this general area because priority is determined by victim probability as well as coverage. As a consequence, new images containing already marked victims may enter the queue even though they represent only minor increases in coverage. Under these conditions, the *cued* display frequently confused operators leading them to mark the same victim twice or even three times at the same location. Augmenting the priority computation by considering whether a target may have been already marked by the operator could alleviate this problem. This poses a new challenge for ATR since it will have to integrate markings placed by the user with its detections.

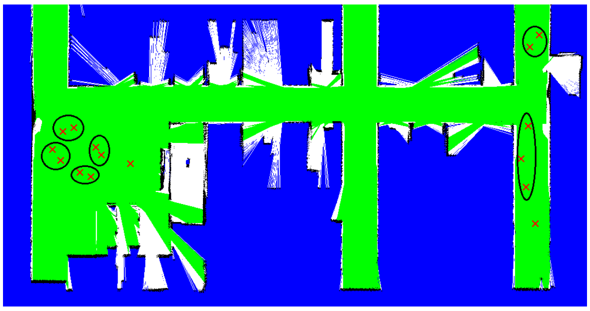

Figure 8. Example for target indication group map

Participants in the streaming video condition were confronted with a bank of videos (Fig. 3), much like a security guard monitoring too many surveillance cameras. Informal observation of participants suggests that due to the frequent distractions of robot operation, victims appearing and disappearing from view, and the need to switch back and forth between tasks, the operator puts a great deal of effort into task allocation and feels intense time pressure. While we undertook this study to determine whether asynchronous video might prove beneficial to larger teams, we found performance to be essentially equivalent to the use of streaming video but with lower errors and workload.

Suitability for multi-operator control is another potential advantage for asynchronous displays such as the image queue. Operators attempting to control or monitor robot teams in real time would be faced not only with the daunting task of controlling and coordinating their own robots but with coordinating with others trying to perform the same difficult tasks. Asynchronous control such as the image queue provides convenient ways to divide tasks functionally among operators, such as allocating exploration and target identification to different operators. Shifting focus from platforms and camera video to the network and regions being explored allows searchers to concentrate on their primary search task rather than on driving or monitoring robots. Just as our image queue operators were called upon to teleoperate robots out of trouble from time to time, we envision future systems which are controlled at both network and platform levels. To realize this kind of control architecture, we propose a call center approach in which some operators address independent control needs for monitoring and exploration of UVs, while other operators address independent location-based images in a queue for victim marking and other perceived tasks. Because synchronous control operators must sacrifice a global perspective to maintain local control of platforms and asynchronous operators sacrifice temporal resolution to gain a global perspective losing situational awareness will be one of the major hazards to be addressed.

Because of these concerns, we want to explore the effects of combing heterogeneous levels of control on large robot teams controlled by multiple operators. In these experiments we hope to compare functional allocation, platform-based allocation, and hybrid allocation schemes employing call center, self-selected, and other task assignment regimes.

ACKNOWLEDGMENT

This research has been sponsored in part by AFOSR FA955008-10356 and ONR Grant N0001409-10680

REFERENCES

Chien, S., Wang, H, & Lewis, M. (2010). Human vs. algorithmic path planning for search and rescue by robot teams, *Proceedings of the 54th Annual Meeting of the Human Factors and Ergonomics Society* (HFES'10), 379-383.

Cooke, N., Pringle, H., Pedersen, H. and Connor, O. (Ed.) (2006) *Human Factors of Remotely Operated Vehicles*. Amsterdam, NL: Elsevier.

Crandall, J., Goodrich, M., Olsen, D. and Nielsen, C. (2005) Validating human-robot interaction schemes in multitasking environments. *Proceedings of IEEE Transactions on Systems, Man, and Cybernetics*, Part A, 35(4), 438–449.

Hart, S., and Staveland, L. (1998) Development of a multi-dimensional workload rating scale: Results of empirical and theoretical research. *In P. A. Hancock & N. Meshkati (Eds.), Human mental workload*, 139-183. Amsterdam, The Netherlands: Elsevier

J. McLurkin, J. Smith, J. Frankel, D. Sotkowitz, D. Blau, B. Schmidt. (2006) Speaking swarmish: Human-robot interface design for large swarms of autonomous mobile robots, *Proceedings of the AAAI Spring Symposium*, Stanford, CA, USA

Kira, Z. and Potter, M. 2010. Exerting Human Control Over Decentralized Robot Swarms. *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*. IROS'10, 566-571.

Lewis, M., Wang, H., Chien, S., Velagapudi, P., Scerri, P. and Sycara, K. (2010) Choosing autonomy modes for multirobot search, *Human Factors, 52(2)*, 225-233.

Miller, C. and Parasuraman, R. (2007) Designing for flexible interaction between humans and automation: Delegation interfaces for supervisory control, *Human Factors, 49(1)*, 57-75.

Olsen, D. and Wood, S. 2004. Fan-out: Measuring Human Control of Multiple Robots. *In Proceedings of Human Factors in Computing Systems (CHI'04)*, Vienna, Austria: ACM Press, 231-238.

Pepper, C., Balakirsky, S., Scrapper, C. (2007) Robot Simulation Physics Validation. *Proceedings of PerMIS'07*

Robocup Rescue VR. (2010) http://www.robocuprescue.org/wiki/index.php
?title=VRCompetitions#Singapore_2010

Scerri, P., Liao, E., Lai, L., Sycara, K., Xu, Y. and Lewis, M. (2005) Coordinating very large groups of wide area search munitions. *Theory and Algorithms for Cooperative Systems, World Scientific,* 451-480.

Taylor, B., Balakirsky, S., Messina, E., Quinn, R. (2007) Design and Validation of a Whegs Robot in USARSim. *Proceedings of PerMIS'07*

Velagapudi, P., Wang, H., Scerri, P., Lewis, M., Sycara, K. (2009) Scaling effects for streaming video vs. static panorama in multirobot search. *IEEE/RSJ International Conference on Intelligent Robots and Systems.*

Velagapudi, P., Wang, J., Wang, H., Scerri, P., Lewis, M., and Sycara, K. (2008) Synchronous vs. Asynchronous Video in Multi-Robot Search. *Proceedings of first International Conference on Advances in Computer-Human Interaction.* ACHI'08