Scalable Target Detection for Large Robot Teams

Huadong Wang, Andreas Kolling, Shafiq Abedin,
Pei-ju Lee, Shih-Yi Chien, Michael Lewis
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA 15260 U.S.A.
huw16@pitt.edu, akolling@pitt.edu, sha33@pitt.edu,
pel30@pitt.edu, shc56@pitt.edu, ml@sis.pitt.edu

Nathan Brooks, Sean Owens,
Paul Scerri, Katia Sycara
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213 U.S.A.
nbb@andrew.cmu.edu, owens@cs.cmu.edu,
pscerri@cs.cmu.edu, katia@cs.cmu.edu

ABSTRACT

In this paper, we present an asynchronous display method, coined *image queue*, which allows operators to search through a large amount of data gathered by autonomous robot teams. We discuss and investigate the advantages of an asynchronous display for foraging tasks with emphasis on Urban Search and Rescue. The *image queue* approach mines video data to present the operator with a relevant and comprehensive view of the environment in order to identify targets of interest such as injured victims. It fills the gap for comprehensive and scalable displays to obtain a network-centric perspective for UGVs. We compared the *image queue* to a traditional synchronous display with live video feeds and found that the *image queue* reduces errors and operator's workload. Furthermore, it disentangles target detection from concurrent system operations and enables a call center approach to target detection. With such an approach we can scale up to very large multi-robot systems gathering huge amounts of data that is then distributed to multiple operators.

Categories and Subject Descriptors

I.2.9 [**Artificial Intelligence**]: Robotics – *operator interfaces.*

General Terms

Human Factors, Experimentation, Performance, Algorithms

Keywords

Human-robot interaction, metrics, evaluation, multi-robot system

1. INTRODUCTION

Many applications such as interplanetary construction, search and rescue in dangerous environments, or cooperating uninhabited aerial vehicles have been proposed for multi-robot systems (MrS). Controlling these robot teams has been a primary concern of many HRI researchers. These efforts have included theoretical and applied development of the Neglect Tolerance model and Fan-out model to characterize the control of independently operating robots [1, 2], predefined rules to coordinate cooperating robots as in Playbook™ [3] and Machinetta [4], and techniques for influencing teams obeying biologically inspired control laws [5, 6, 7]. While our efforts to increase span of control over unmanned vehicle (UV) teams appear to be making progress, the asymmetry between what we can command and what we can comprehend is growing. 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 expressly to monitor and interpret video being gathered by UVs. This can be a difficult task for even a single camera [8] and begins exceeding operator capability before reaching ten cameras [9, 10]. With increasing autonomy of robot teams and plans for biologically inspired swarms of much greater size the problem of absorbing and benefiting from their product seems even more pressing than learning how to command them.

One approach to this problem is the Picture-in-Picture display (PiP), a specialized solution for integrating unmanned aerial vehicle (UAV) camera video [11, 12]. In a PiP display current video, scaled and transformed to reflect the position and orientation of the UAV and camera, is projected onto a map of the area being surveyed. As the cameras move about, the areas of the map falling under their view are refreshed. By presenting camera views in context, the PiP display eliminates many of the mental transformations and confusions necessary to interpret video being viewed from a remote camera [13]. Because the operator can attend to the information being gathered by the team of UAVs rather than the video from any particular UAV, the PiP display provides a network-centric [14] view of the collected information and has been shown to lead to enhanced situation awareness [11]. Another favorable property of the PiP display is that the ease of use should be independent of team size because the size and resolution of the map remains constant. As additional UAVs are added, regions of the map will be updated more frequently reducing uncertainty but without imposing additional load on the operator. In fact, as the update rate is increased the operator will no longer have to predict or extrapolate and can simply observe making the task even easier.

The integration of unmanned ground vehicles (UGVs) into a network-centric view and reducing the difficulty of monitoring their video is a more challenging problem because there is no convenient correlate of an aerially viewed map to provide a common representation. UGV camera views typically provide a projection of a 3D area onto a horizontal plane, the camera plane of the UGV. The resulting image captures the space between a UGV and obstructions such as walls or vegetation. Even though the view may be geo referenced, how informative it is will depend on the degree to which it is relatively unobstructed. Differences in geometry also make UGV views much less informative. Because

HRI'11, March 6–9, 2011, Lausanne, Switzerland.
Copyright 2011 ACM 978-1-4503-0561-7/11/03...$10.00.

a UAV views the scene below it from a relatively long constant distance, objects and surroundings, obeying the inverse square law, have spatial relations that appear relatively constant. For a UGV, by contrast, objects will remain within camera view for long periods changing drastically in size and appearance as the UGV approaches and moves away from them. In this case views need to be chosen carefully for informativeness rather than simply transformed to fit a map.

While PiP views are inherently spatially organized and present collected data in its entirety this may not necessarily work well for UGVs. Because UGV cameras collect many drastically different views of the same scene there is no simple algorithmic way to fuse all these views simultaneously into a single fully informative display. Simple projections onto a map as for PiP are impossible because such projections would involve a complete 3d reconstruction of the environment. Even if this were achieved, with advances in 3d mapping and 3d reconstruction from images, viewing such a 3d reconstruction will also not provide a simultaneous perspective on the entire environment. Once simultaneity is lost, the user is no longer guaranteed to be able to see new events on screen as they occur. But an asynchronous display of data may suffice for static environments such as presumed for Urban Search and Rescue (USAR) with immobilized victims and may prove valuable also without a 3d reconstruction.

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 UGVs. We will present one particular design for such a display and test it in the context of USAR with large robot teams with some degree of autonomy and supervised by a single operator.

2. FORAGING AND ASYCHRONOUS DISPLAY OF INFORMATION

Foraging tasks, carried out with a large robot team, require an exploration that needs to be more than simply moving each robot to different locations in the environment. Acquiring a specific viewpoint of targets of interest, e.g. victims in a disaster scenario, is of greater concern and increasing the explored area is merely a means to this end. While a great deal of progress has been made for autonomous exploration the identification of targets is still done by human operators who ensure that the area covered by robots has in fact been thoroughly searched for the desired targets. Without means to combine the data gathered by all robots the human operator is required to synchronously monitor their output, such as a video feed for each robot. This requirement and load on the human operator directly conflicts 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 [20, 21] confirmed that search performance on these tasks is directly related to the frequency with which the operator shifts attention between robots, possibly due to targets missed in the video stream while servicing other robots.

An asynchronous display method can alleviate the concurrent load put on the human operator and disentangle the dependency of tasks that require the video feed. Furthermore, it can avoid attentive sampling among cameras by integrating multiple data streams into a comprehensive display. This in turn allows the addition of new data streams without increasing the complexity of the display itself. A first approach for an asynchronous display is explored in [28]. The method therein is motivated by asynchronous control techniques previously used in extraterrestrial NASA applications. These are faced with limited bandwidth and communication lags due to large interplanetary distances. Such lags make a direct operation and viewing of all outputs impossible. Instead, the robot team is instructed to gather information only at specific locations in the form of panorama images from an omnidirectional overhead camera. The operator then searches through all panorama images and determines the location of specific targets. The approach was tested in experiments which compared performance for operators controlling four robots in a team using streaming or asynchronous panorama displays. No significant improvement was found in terms of found victims, but the frequency of shifting focus between robots was correlated with performance for streaming video but not for asynchronous panoramas. As expected, the asynchronous display of information alleviates the need for excessive switching. It was conjectured that with larger robot teams the benefit of avoiding attention switching will increase and have an effect on performance in terms of victims. Further experiments in [29] scaling the team size to eight and twelve robots, however, found no further significant improvements. But this approach did not utilize all the available data from the video feeds that robots gather, so a huge amount of potentially useful information was discarded for the panorama condition. Furthermore, the operator needs to give the robots additional instructions where to sample panoramas.

In contrast to previous work our approach in this paper allows the use of autonomous exploration and we present an asynchronous display that mines all of the robot video feeds for relevant imagery. This imagery is then given to the operator for analysis. We coin this type of asynchronous display *image queue* and compare it to the traditional synchronous method of streaming live video from each robot (*streaming video*). In the next section we describe our test bed and followed by a detailed description of the *image queue* and a comparison with *streaming video.*

3. METHODS

3.1 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 urban search and rescue (USAR) robots and environments 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 uses Epic Games' UnrealEngine3 [15] to provide a high fidelity simulator at low cost and also serves as the basis for the Virtual Robots Competition of the RoboCup Rescue League. Other sensors including sonar and audio are also accurately modeled. Validation data showing close agreement in detection of walls and associated Hough transforms for a simulated Hokuyo laser range finder are described in [17]. The current UnrealEngine3 integrates Nvidia's PhysX physics engine [18] to support high fidelity rigid body simulation. Validation studies showing close agreement in behavior between USARSim models and real robots being modeled are reported in [19, 20, 21, 22, 23] as well as agreement for a variety of feature extraction techniques between USARSim images and camera video are reported in Carpin et al. [16].

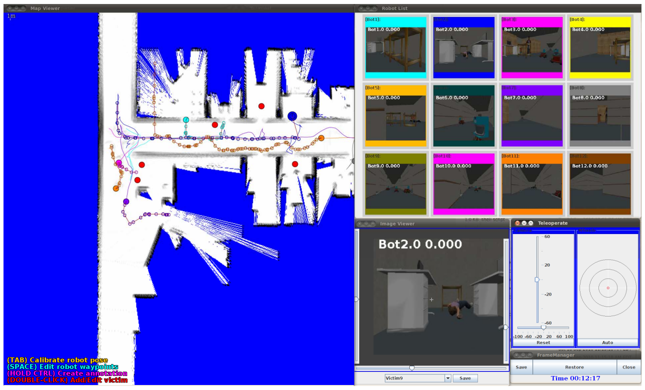


Figure 1. GUI for the *streaming video* condition.

MrCS (Multi-robot Control System), a multi-robot communications and control infrastructure with accompanying user interface, developed for experiments in multirobot control and RoboCup competition [24] was used in this experiment. MrCS provides facilities for starting and controlling robots in the simulation, displaying multiple camera, and laser output, as well as maps, and supporting inter-robot communication through Machinetta, a distributed multi-agent coordination infrastructure. Figure 1 shows the elements of the conventional GUI for the *streaming video* condition. The operator selects the robot to be controlled from the colored thumbnails at the top right of the screen. These thumbnails also show the live video streams from all robots. To view more of the selected scene shown in the large video window the operator uses pan/tilt sliders to control the camera of the selected robot. The current locations and paths of the robots are shown on the Map Viewer (bottom left). 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). Apart from the interface for displaying relevant information and allowing controlling robots manually MrCS also contains a number of autonomy functions that are described in more detail Section 3.3. In the experimental system an image queue display replaces the live video feeds. The ability to select a robot from the map to teleoperate and view streaming video remains

3.2 Image Queue Interface

The goal of the *image queue* interface is to best utilize the advantages of an asynchronous display and to maximize the amount of time human operators can spend on tasks which human's performs better than robots. Currently, for USAR, this is the case for tasks such as victim identification and navigating robots out of dangerous areas in which they got 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 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*.

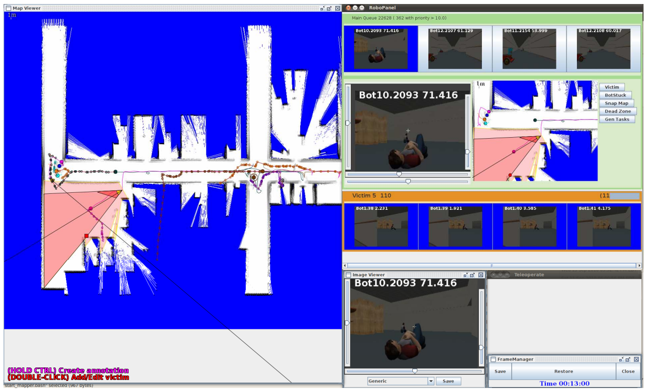


Figure 2. GUI for the *image queue* conditon.

The *image queue* interface (Figure 2) 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. The filter attempts to reduce redundancy by only showing highly relevant images from the video stream. Relevance is scored by computing a utility for every image that determines its priority in the queue displayed in the filmstrip viewer. To achieve this we store every frame from all video streams in a database together with associated robot poses and laser scans taken at the time of capture. From this database we can retrieve any image and compute its utility. The computation of utility can be adapted to a particular application and for our experiment we computed utility via the area covered seen in an image. This visual coverage is computed by referencing the image in the map as seen in Figure 2 and 3. Images with larger areas receive higher utility scores. Areas that have already been seen by other images in the filmstrip viewer do not count towards utility. In colloquial terms this kind of utility picks images that cover large areas with minimal overlap. Figure 3 illustrates this concept of utility with a simple example while Figure 4 presents an overview of the steps involved in this process.

More precisely, we compute a utility $u(I)$ for every image I as follows. Let $F \subset \Re^2$ be the workspace of the robots, i.e. the free space in which the can navigate. F is uncovered by the robots as they explore and build a map. Let S be all images already seen and processed by the user from the image queue and let D be all images stored in the database. Initially S is empty and D grows as robots explore. Every image I in D has an associated area $a(I) \subset F$ which is the area visible in the image. This area can be computed by referencing the image in the map of F. The relevant area for an image I is

$$\tilde{a}(I)= a(I) \setminus \bigcup_{I' \in S} a(I')$$

i.e. all yet unseen area. Now we define $u(I) = area(\tilde{a}(I))$ which become the priority according to which we sort our images in D. Amongst these we present all those with the highest priority.

Obviously, once a user sees and processes an image the set S changes the priorities have to be updated. To avoid frequent updates we can compute the utility for I supposing that all higher priority images are already in S. Then the utility of an image only updates when new imagery is added to the database D at a location nearby.

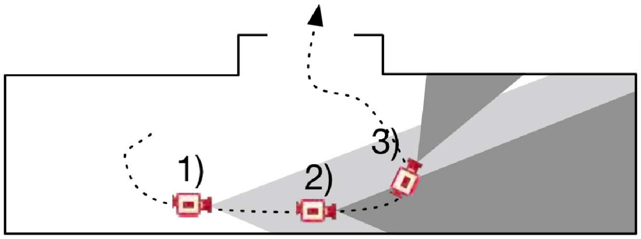


Figure 3. An illustration of the utility of individual frames from a video stream. The frame taken at 1) has the largest visual coverage and highest utility while the frame at 2) has no utility since it is entirely overlapped by 1). Frame 3) has some utility since it provides coverage in an area not covered by 1).

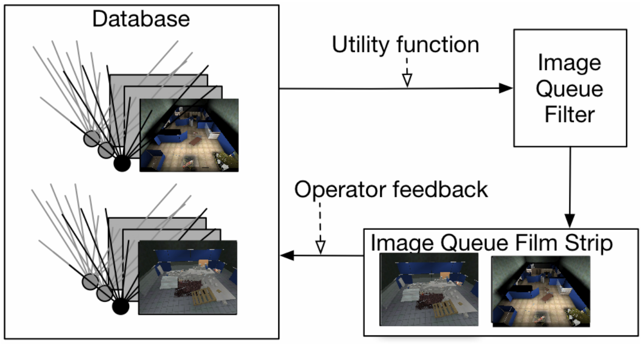
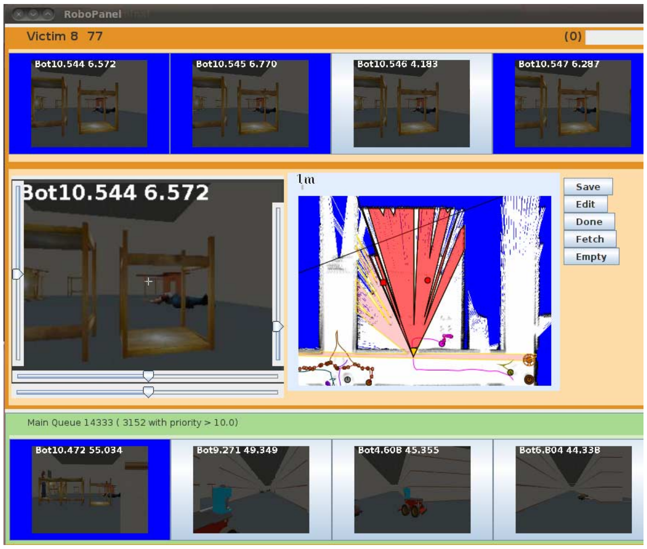


Figure 4. An illustration of the system architecture for the *image queue*. A database stores observations made by multiple robots such as laser scans and video frames. From these a filter extracts high utility images which are then presented for analysis in a film strip.

By aggregating imagery with the highest utility 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. Notice that exploration can continue while operators view the *image queue* so long as robots are sufficiently autonomous (or controlled by other operators). Operators can either click through 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% within the first 100 frames.

For every image in the queue the operator can mark victims in the area covered by the image and request additional information on a location. Such a request opens a sub-queue (Figure 5) that contains images retrieved from the database taken from nearby positions. This sub-queue helps the operator to localize the victim properly to find better perspectives to view the victim. Once a victim is identified, the operator can provide information regarding the status of the victim with the help of the viewer and mini map. This feature provides some amount of contextual information but in contrast to the *streaming video* does not all the user to select the robot that took this imagery and drive it to desired positions since it may well have travelled to further locations already.



Main Queue 14333 (3152 with priority > 10.0)

Figure 5. Sub-queue for victims in the *image queue* GUI.

3.3 System Autonomy

In previous experiments [10] it was demonstrated that a high degree of autonomy can improve performance in foraging tasks, especially in search and rescue scenarios. Consequently, in the current version of MrCS we included further autonomous functions while also improving on existing capabilities. A new Segment Voronoi Diagram (SVD) path planner replaced the random tree planner used in earlier studies. The new planner generates paths that maintain a safe distance to nearby obstacles. Such paths are generally longer, smoother and more human-like. In [25] it was shown that operators are able to also follow less smooth paths rather well, but the added safety of the paths also benefits autonomous navigation. The use of a path planner and autonomous navigation to drive the robots is one major distinction to the panorama study [28] for which paths were generated manually by participants to reach specified panorama locations.

To autonomously guide the exploration process and further relief the operator we used a frontier-based exploration approach. From the map our algorithm identifies regions on the boundary between known and unknown space and ranks these according to expected benefit of visiting them. Each robot identifies these locations locally within their own map, created from each robots history of laser scans, and communicates these regions to other robots. Machinetta is used coordinate the assignment of exploration locations to robots. Then paths are generated for every robot to visit its assigned locations to gather further laser and video data, improve the map, and then repeat the cycle by computing new exploration locations. Figure 6 shows a few of the structures used and computed by the autonomy.

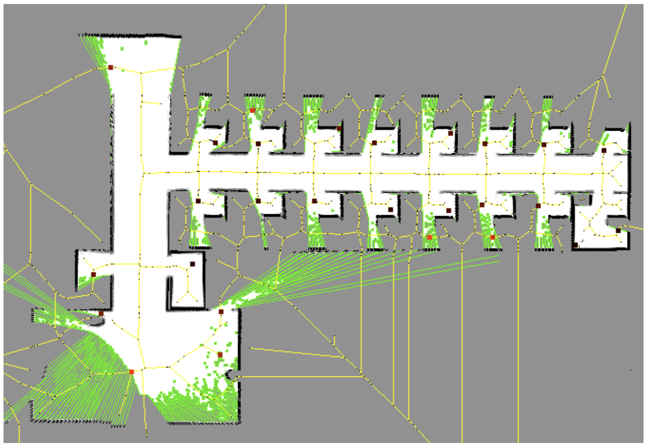


Figure 6. A sample map (obstacles in black), path planner roadmap (yellow), frontier cells (green), and locations for exploration (black and red squares). Preferred exploration locations are red.

3.4 Experimental Conditions

A large USAR environment previously used in the 2010 RoboCup Rescue Virtual Robots competition [26] was selected for use in the experiment. The environment was an office like hall with many rooms and full of obstacles like chairs, desks, and bricks. Victims were evenly distributed within the environment. Robots can enter the environment from one of two possible entrances. 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. On average participants explored only one fifth of the environment finding 8.8 victims.

The experiment followed a two condition repeated measures designs comparing the conventional MrCS displays (*streaming video*) with MrCS augmented by the experimental *image queue* display counterbalancing conditions and entrance points for robots. Automated path planning to improve search performance and autonomous exploration was used in both conditions. Because the laser map is built up slowly as the environment is explored and the office like environment provides few distinctive landmarks there was little opportunity for participants to benefit from prior exposure to the environment from a different entrance location.

The operators performed a supervisory control task in which the robots navigated autonomously with the operator allowed to override by directing them through new waypoints. When necessary, participants were able to teleoperate the in-focus robot to extricate it when it became stuck.

3.5 Participants

32 paid participants were recruited from the University of Pittsburgh community balanced among conditions for gender. None had prior experience with robot control although most were frequent computer users.

3.6 Procedure

After providing demographic data and completing a perspective taking test participants read 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* condition for 10 min each. Participants were encouraged to find and mark at least one victim in the training environment under the guidance of the experimenter. 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 using either the *streaming video* or *image queue* display with a counterbalanced design. At the conclusion of each real task session, participants were asked to complete the NASA-TLX workload survey [27].

4. RESULTS

Data were analyzed using a repeated measures ANOVA comparing *streaming video* with the *image queue* condition. Overall, in both conditions participants were successful in searching through the environment. On average participants in the *streaming video* condition found 9.10 victims while those in the *image queue* condition found 8.51 (Figure 7) without a significant difference between conditions ($F_{1,28}$ = .733, p = .387). The area explored (Figure 7) for both conditions did also not differ significantly ($F_{1,28}$ = 2.147, p = .154).

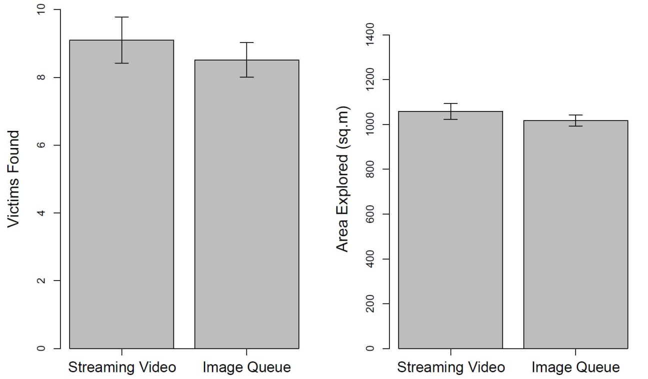


Figure 7. Victims found and area explored.

Every mark a participant made for a victim was compared to ground truth to determine whether there was in fact 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*.

While we initially tried assigning marks to the nearest robot it led to assigning some marks at distances that did not seem credible. Using longer distances also made anomalies such as assigning a mark to a victim in another room more likely. The fact that the data show low rates of false positives (marking a victim beyond the 2 m radius) of between 1 (image queue) and 2.4 (streaming video) while successful markings were 8.5 and 9.1 suggests that the 2 m criterion is doing well at associating most marks with victims. Visual inspection confirms this with most false positive marks falling well outside the 2 m radii rather than nearby.

For both types of errors the data showed a significant advantage for the *image queue* condition (Figure 8). There were significantly fewer false positives ($F_{1,28}$ = 13.032, p = .001) as well as fewer false negatives ($F_{1,28}$ = 5.526, p = .026) with an average in the *image queue* condition of 7.48 while participants in the *streaming video* condition missed 9.34 victims on average (Figure 8).

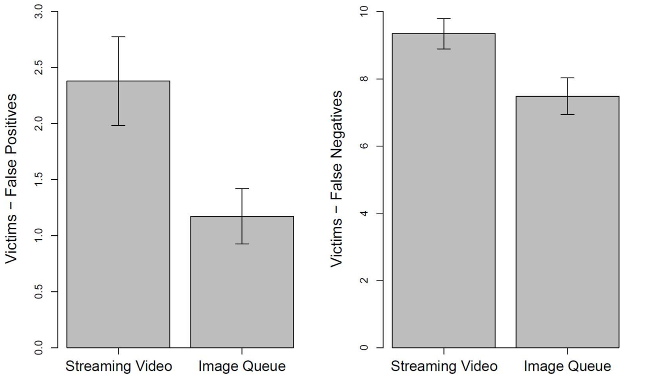


Figure 8. Marking errors of victims.

In addition we monitored the number of instances that participants initiated a teleoperation session with an individual robot. The repeated measures ANOVA shows a significant difference for teleoperation instances between the *streaming video* and *image queue* condition (Figure 9). Participants in the *streaming video* condition teleoperated on average 21.24 times while participants in the *streaming video* conditon teleoperated on average 4.97 times ($F_{1,28}$ = 150.719, p < .001).

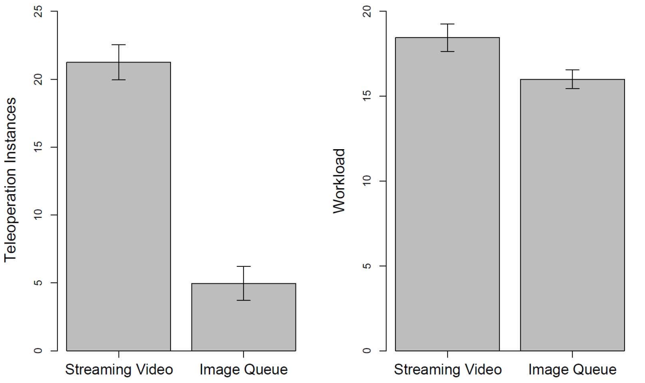


Figure 9. Teleoperation and workload.

The full scale NASA-TLX workload measure also revealed a significant difference, which is unlike the earlier studies [28] where no advantage was found for panorama pictures GUI. A significant advantage in workload ($F_{1,28}$ = 7.347, p = .001) was observed favoring the *image queue* condition (Figure 9). Examining individual dimension of workload comparing streaming video and image queue we found significant differences for temporal demand ($F_{1,28}$ = 6.503, p = .016) and effort ($F_{1,28}$ = 4.576, p = .040) (Figure 10).

5. DISCUSSION AND CONCLUSION

The purpose of our experiment was to examine the impact of our asynchronous *image queue* display on overall performance. It presents information to subjects asynchronously but ordered by a quality metric that relates to the utility of the information. This stands in contrast to the video stream that presents information as it becomes available. Either method can have advantages and disadvantages. Especially navigation tasks benefits from immersion and situation awareness. In fact, in the *streaming video* condition we observed more teleoperation instances while participants for the image queue avoided teleoperating the robots and relied more heavily on the autonomy. Consequently, one would expect them to benefit from further improvements to the autonomy for which there is ample room. As autonomy improves, we ultimately see the need for navigation reduced to situations in which the operator has to assist robots to fix 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 to be completed entirely asynchronously without any penalties for performance in terms of the number of victims. Even more so, by decoupling it we reduced the number of errors that occur for marking victims, false positives and negatives alike, while at the same time reducing overall workload. We can hence conclude that the added situation awareness from teleoperating robots while searching for victims provides little to no benefit for such a system with many robots. On the other hand, the reduction in errors for *image queue* is particularly significant because avoiding missed targets is crucial to most foraging tasks. Especially for USAR thoroughness and correctness are two of the most important performance metrics.

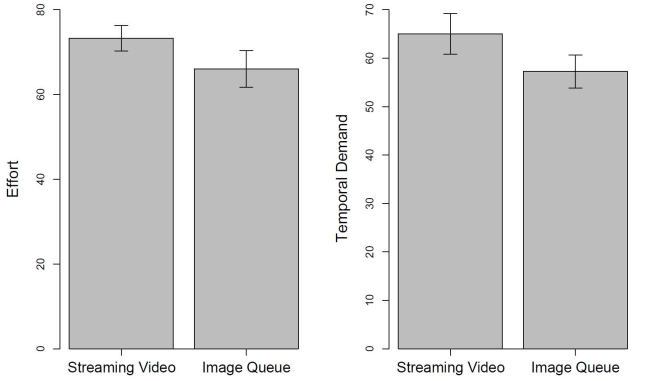


Figure 10. Effort and temporal demand.

Participants in the *streaming video* condition were confronted with a bank of videos much like a security guard monitoring many surveillance cameras. Informal observation of participants suggests that the continued monitoring of video feeds synchronously during system operation puts a greater effort on operators. Not only does it require continuous attention but participants switch between tasks more often (whenever a victim appears in a video feed). This leads to an impression of time pressure that may well contribute to the increased number of errors. The significant difference of individual dimension of workload, temporal demand and effort, supports this observation.

The overall significance of the successful decoupling of detection from other tasks goes further than the mere reduction of errors, workload, and teleoperation instances. It allows the design of a system that treats target detection tasks as notifications for a call center. Regardless of the origin of the notification our experiment indicates that we could assign an image from the *image queue* to any operator and hence scale our operations in terms of the number of operators. The small amount of contextual information in form of a mini map and some additional imagery from the database seems to suffice and does not jeopardize performance. As autonomous exploration continues to improve and scales to very large teams of autonomous robots, we can now ensure that our current methods for target detection by human operators scale with it.

There is, however, one problem with this approach. Errors that originate from a complex interaction between tasks will be difficult to detect by human operators. Effectively, operators rely more on the autonomy and system design and sacrifice the ability to have a complete overview of the system and its functions. Such a sacrifice is often made when scaling large and complex systems but its impact on foraging tasks is not yet well understood. We conjecture that it is in fact less dramatic than a failure in a complex plant monitoring operation which can lead to precisely the type of disasters that our robot teams are supposed to find victims in.

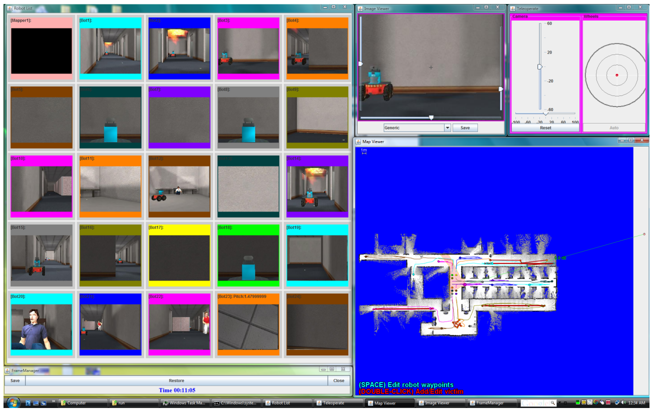


Figure 11. 24 robots in the *streaming video* mode.

At this point we can well imagine two operators supervising 24 UVs and responding to UV originated requests such as verifying/marking targets as well as other types of alarms that may require them to regain situation awareness beyond what the *image queue* requires, such as getting stuck UVs out of trouble (Figure 11). As further work we intend to directly address scalability with a call center approach and investigate its effects on even larger teams with multiple operators. Another useful extension of this work to moving targets is also planned. Evidently, an asynchronous display method will face particular challenges when dealing with dynamic environments and targets and will require more sophisticated techniques.

6. ACKNOWLEDGMENTS

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

7. REFERENCES

[1] 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.

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

[3] 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.

[4] 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.

[5] 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.

[6] McLurkin, J., Smith, J., Frankel, J., Sotkowitz, D., Blau, D. and Schmidt, B. 2006. Speaking Swarmish: Human-Robot Interface Design for Large Swarms of Autonomous Mobile Robots. In *Proceedings of AAAI Spring Symposium*.

[7] Ding, X., Powers, M., Egerstedt, M., Young, S. and Balch, T. 2009. Executive Decision Support: Single-Agent Control of Multiple UAVs. *IEEE Robotics & Automation Magazine*, 16 (2), 73-81.

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

[9] Humphrey, C., Henk, C., Sewell, G., Williams, B. and Adams, J. 2007. "Assessing the Scalability of a Multiple Robot Interface," In *Proceedings of the 2nd ACM/IEEE International Conference on Human-Robotic Interaction (HRI'07),* 239-246.

[10] 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.

[11] Draper, M.H., Calhoun, G.L., Nelson, J.T. and Ruff, H.A. 2006. Evaluation of synthetic vision overlay concepts for UAV sensor operations: landmark cues and picture-in-picture. In *Proceedings of the Human Factors and Ergonomic Society 50th Annual Meeting*, 121-125.

[12] Hunn, B. 2005. The Human Challenges of Command and Control with Multiple Unmanned Aerial Vehicles. In *Proceedings of the Human Factors and Ergonomics Society 49th Annual Meeting*, 20-24(5).

[13] Gugerty, L. and Brooks, J. 2001. Seeing Where You Are Heading: Integrating Environmental and Egocentric Reference Frames in Cardinal Direction Judgements. *Journal of Experimental Psychology: Applied*, 7 (3), 251-66.

[14] Alberts, D., Garstka, J. and Stein, F. 1999. *Network-centric Warfare: Developing and leveraging information superiority*, the Command and Control Research Program.

[15] UE3 http://www.unreal.com/technology.php, accessed September 15, 2010.

[16] Carpin, S., Stoyanov, T., Nevatia, Y., Lewis, M. and Wang, J. 2006. Quantitative assessments of USARSim accuracy. In *Proceedings of PerMIS*.

[17] Carpin, S., Wang, J., Lewis, M., Birk, A., and Jacoff, A. 2006. High fidelity tools for rescue robotics: Results and perspectives, *Robocup 2005 :Robot Soccer World Cup IX,* Springer LNAI vol 4020, Springer, 301-311.

[18] PhysX, http://www.nvidia.com/object/physx_new.html, accessed September 15, 2010.

[19] Carpin, S., Lewis, M., Wang, J., Balakirsky, S. and Scrapper, C. 2006b. Bridging the gap between simulation and reality in urban search and rescue. *Robocup 2006: Robot Soccer World*

Cup X, Springer, Lecture Notes in Artificial Intelligence, 1-12.

[20] Lewis, M., Hughes, S., Wang, J., Koes, M. and Carpin, S. 2005. Validating USARsim for use in HRI research, In *Proceedings of the 49th Annual Meeting of the Human Factors and Ergonomics Society*, Orlando, FL, 2005, 457-461.

[21] Pepper, C., Balakirsky, S. and Scrapper, C. 2007. Robot Simulation Physics Validation, In *Proceedings of PerMIS'07*.

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

[23] Zaratti, M., Fratarcangeli, M and Iocchi, L. 2006. A 3D Simulator of Multiple Legged Robots based on USARSim. *Robocup 2006: Robot Soccer World Cup X, Springer, LNAI*, 13-24.

[24] Balakirsky, S., Carpin, S., Kleiner, A., Lewis, M., Visser, A., Wang, J., and Zipara, V. 2007. Toward hetereogeneous robot teams for disaster mitigation: Results and performance metrics. *RoboCup Rescue, Journal of Field Robotics,* 24(11-12), 943-967.

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

[26] Robocup Rescue VR http://www.robocuprescue.org /wiki/index.php?title=VRCompetitions#Singapore_2010, accessed September 15, 2010

[27] Hart, S., and Staveland, L. 1988. Development of a multi-dimensional workload rating scale: Results of empirical and theoretical research. In P. A. Hancock & N. Meshkati (Eds.), *Human mental workload*, Amsterdam, The Netherlands: Elsevier, 139-183.

[28] Velagapudi, P., Wang, J., Wang, H., Scerri, P., Lewis, M., and Sycara, K. 2008. Synchronous vs. Asynchronous Video in Multi-Robot Search, In *Proceedings of first International Conference on Advances in Computer-Human Interaction.* (ACHI'08), 224-229.

[29] Velagapudi, P., Wang, H., Scerri, P., Lewis, M. and Sycara, K. 2009. Scaling effects for streaming video vs. static panorama in multirobot search, In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems.* IROS'09, 5874-5879.