Attaining Situational Awareness for Sliding Autonomy

Brennan P. Sellner
Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA USA
bsellner@andrew.cmu.edu

Laura M. Hiatt
Computer Science Department
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA USA
lahiatt@cs.cmu.edu

Reid Simmons, Sanjiv Singh
Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA USA
reids@cs.cmu.edu, ssingh@cs.cmu.edu

ABSTRACT
We are interested in the problem of a human operator who has to respond to requests for help from an autonomous robotic construction team. A difficult aspect of this problem is gaining an awareness of the requesting robot's situation quickly enough to avoid slowing the whole team down. One approach to speeding the initial acquisition of situational awareness is to maintain a buffer of data, and play it back for the human when their help is needed. The paper reports on an experiment to determine the proper composition and length of this buffer for our domain of multi-robot construction. The experiments show that 5 - 10 seconds of one raw video feed in combination with a processed display led to the fastest operator attainment of situational awareness. We draw several conclusions from this experiment, which may generalize to other scenarios.

Categories and Subject Descriptors
I.2.9 [**Artificial Intelligence**]: Robotics—*Operator interfaces*; H.1.2 [**Models And Principles**]: User/Machine Systems—*Human factors*

General Terms
Experimentation, Human Factors

Keywords
Situational Awareness, User Study, Sliding Autonomy, Case Study

1. INTRODUCTION
"Everything fails sometimes" is an axiom in the robotics community. Given finite resources, it is nearly impossible to create a robotic system that can operate in an open, dynamic environment with zero failures. More importantly, there are always unexpected error conditions, which the robot's programmers were unable to anticipate. In such cases, a purely autonomous system is left with little recourse: it can attempt a few generalized recovery strategies, but there will always be corner cases that result in terminal failure.

Human Robot Interaction '06 Salt Lake City, Utah USA

Rather than attempting to incrementally account for every possible error condition, and accept a terminal failure if we encounter an unexpected case, we believe that autonomous robots or teams of robots should instead be designed to handle common failures and merely recognize, at least in a general sense, rare failures. Most failures can be detected at some level, such as exceeding a limit, but sufficient information for automated recovery is often not available.

Simply detecting errors is of little use to the autonomous system without a way to translate the error into a recovery method. Since it's impractical to do so in code for all conceivable sets of recovery methods and errors, we solve this dilemma by involving human(s) in the team. When a robot believes it is in trouble, but has no way of recovering on its own, it may request help from a human team member. This takes advantage of the strengths of the autonomous system, such as its ability to quickly and accurately perform routine, repetitive, or long-duration tasks, while using the flexibility of the human partner on an as-needed basis to fill in the gaps in the autonomous system.

Our domain is multi-robot construction, which is rife with opportunities for coordination and collaboration between robots and humans. The current scenario involves three heterogeneous robots working together to assemble a square structure out of four beams and four connecting nodes. The nodes, mounted on wheeled bases, must be braced before a beam may be connected. This results in three natural roles within the team: a bracing robot, a robot to perform the fine manipulation necessary to connect a beam to a node, and a third to provide a mobile sensor platform (Figure 2).

Within our construction domain, a human teleoperator is available to help the system as needed. In our previous work [4], we compared pure teleoperation, pure autonomy, and two mixed (Sliding Autonomy) approaches. The two mixed approaches varied as to whether or not the human was allowed to constantly monitor the system's progress and proactively assume control. In this paper, we investigate the case in which the human is occasionally asked to provide assistance while performing other unrelated tasks. The challenge for the human in such a scenario is to swiftly attain situational awareness. By situational awareness, we mean an understanding of: the robotic team's and workspace's current state, how far the team has progressed in the assembly process, what caused the team to ask for help, and what is an appropriate next action. There are clearly degrees of situational awareness: knowing that the robot is in Alaska is a coarser degree of knowledge about the robot's state than realizing that it is 1.12 meters north of Beam 4.

For the purposes of this paper, we define situational awareness as sufficient understanding of the team's state to formulate a short-term plan of action. This will vary somewhat on a case-by-case basis, but in general will only involve comprehension of the spatial relationships between the small set of objects and robots directly involved in the current task.

Attaining situational awareness is nontrivial, as the human has not been monitoring the team's progress while fulfilling their other responsibilities. It is often quite difficult to determine the problem which triggered the request for help by simply observing the team's current (static) state. For instance, in a high-clutter environment the cause of the request for help may be ambiguous due to limited camera angles and a lack of depth perception. While moving the robots could help to remove this ambiguity, this may not be safe if they are in close proximity to obstacles. One approach to helping the human safely attain situational awareness is to provide a playback of data for some amount of time preceding the request for help - in other words, maintain a buffer of the system's recent activity and display it to the human when help is needed. This helps remove some of the scene's ambiguity, and also provides information about the team's recent actions, which may further help in determining the current problem.

We have conducted an experiment in order to investigate how the length of this buffer and different combinations of camera angles and synthesized views affect the human's acquisition of situational awareness within our scenario. The experiment showed that 5 - 10 seconds of one raw video feed, plus one synthesized view, yields the best interface for users in our system to gain situational awareness. In addition, we draw a few conclusions which may generalize to other Sliding Autonomy scenarios.

2. RELATED WORK

There has been a significant amount of work over the years on helping humans maintain situational awareness in a number of different scenarios [12]. Much initial work focused on maintaining the situational awareness of pilots [9] [3], while more recent research has investigated the maintenance of situational awareness during teleoperation of search and rescue robots [12]. The primary focus of the existing situational awareness literature is on maintaining the awareness of an operator who is in continual control (or at least is continually monitoring) a robot or robots. This is in contrast to our domain, in which we are interested in helping the operator repeatedly attain situational awareness without monitoring the system between interaction episodes.

One model of situational awareness that is applicable to our domain is that proposed by Endsley [1] [2]. That model defines three levels of the situational awareness: Level 1, perception of environmental elements; Level 2, comprehension of current situation; and Level 3, projection of future states. Level 1 consists of basic perception of cues: an operator who has achieved Level 1 situational awareness has successfully comprehended the bits and pieces of information available to them. Level 2 situational awareness integrates the data perceived in Level 1 and, once achieved, allows the human to derive task-relevant meaning from the raw data perceived by Level 1. The final stage of situational awareness, Level 3, involves the projection of the future state of the system. Operators who have achieved Level 3 situational awareness are able to predict future system behavior from current events and dynamics perceived in Levels 1 and 2. In our experiment, we attempt to determine how long it takes subjects to attain Level 2 situational awareness when provided with differing sources and amounts of historical information.

Teleoperation systems enhance operator situational awareness in different ways. One approach looks at how the operator should view the workspace. In many systems, there are both external views and views from cameras on the robot(s) themselves. Wang and Milgram [11], however, have tried to limit the mental workload required to reconcile those two views by creating a new type of viewpoint. Their view, called a "tether" view, is a display that is neither external nor robot-oriented; instead, it combines the two by simulating how the scene would look from a kite flying behind the robot in the workspace.

In addition to viewpoint, situational awareness is also improved by studying how spatial information of the workspace should be presented to the operator. Lasswell and Wickens [7] investigated ways to improve information displays for pilots in order to improve taxiway safety and traffic flow. They showed that a 3-dimensional, perspective-view of the workspace reduced lateral tracking errors, but that a 2-dimensional, plan-view of the workspace supported greater taxi speeds. This suggested that by giving the 2D display a wider field of view, operators would be able to get a better feel for their situation as well as benefit from the advantages of a plan-view of the workspace. Presumably, techniques such as this that are used to provide a greater degree of situational awareness could also be applied to the situations we are looking at, in which the operator is attempting to gain, not maintain, awareness of the workspace.

A different area of science that is pertinent to our work is that of cognitive psychology. When presented with multiple visual displays, the question arises of how much information operators can process and remember at a time. The cognitive psychology community has done relevant work studying the limitations of human visual working (short-term) memory [10]. Studies have shown that working memory has severe limitations, and that it can only actively hold a few pieces of information at a time. What exactly limits how much information working memory can hold is still under investigation; possibilities include the number of objects attended to, the number of features those objects have, and so forth. Regardless, this research could be very helpful in the design of operator interfaces, as it is important to provide operators with enough, but not too much, information.

3. SITUATIONAL AWARENESS AND SLIDING AUTONOMY

We now provide a brief overview of the form of Sliding Autonomy we investigate in this paper and discuss our approach to helping the human attain situational awareness. As part of the Sliding Autonomy discussion, we examine how the autonomous component decides when to request help from the human component and how an understanding of a human's ability to quickly attain situational awareness can affect this decision. In order to arrive at an interface which will help the human swiftly achieve situational awareness, we look at the limitations of the current approach and some of the design questions that arise when building an interface for this purpose.

Our current research primarily deals with Sliding Autonomy, which addresses the question of how best to meld the complementary talents of human teleoperators and autonomous control systems via various combinations of relative autonomy. On one end of the autonomy spectrum lies pure teleoperation, in which the human teleoperator is in complete control of every aspect of all robots. In general, teleoperation is quite reliable, but inefficient both in time to task completion (due to limited perception and latency) and workload imposed on the human operator. The other extreme is pure autonomy, in which the robotic team acts on its own, with no human involvement whatsoever. Although pure autonomy is often much faster than teleoperation, it is significantly less robust, especially in dynamic domains where all errors cannot be determined *a priori* [4].

Many robotic systems that allow human involvement are limited to these two opposing modes - teleoperation and pure autonomy - with few, if any, options in between. Our goal, however, is a system which melds the respective benefits of teleoperation and pure autonomy while avoiding their associated shortfalls. In this experiment, we examine a Sliding Autonomy mode in which the autonomous system requests help as needed from a human performing other, unrelated, tasks (referred to as "System Initiative" in [4]).

When operating in this mode, the autonomous system maintains models of both its own past performance and any available human teleoperators' skills. These models allow the autonomous system to decide whether to request help from the human because it believes the human will be more efficient. In addition, a robot may request help if it believes it will be unable to recover from a detected failure. The goal of this mode of operation is to allow the autonomous system to increase its robustness by taking advantage of the human's skills and flexibility at need while not unduly loading them. The net effect is a team which is more reliable than pure autonomy and faster than pure teleoperation. In addition, since the human does not need to constantly monitor the autonomous team, they may attend to other tasks while the autonomous system is operating.

Here, we investigate how to design an operator interface to allow the human teleoperator to swiftly and efficiently acquire situational awareness when transitioning from an unrelated task. Addressing this problem yields two benefits: a more efficient user interface, which allows the human to attain situational awareness more quickly; and a more accurate model of how long this acquisition will take, which allows the autonomous system to make more informed decisions about whether to request assistance with a particular task.

The standard operator interface in this situation is identical to that of a full-time teleoperator: it will reflect the current state of the robots, and provide information about the workspace in a variety of forms. However, such an approach suffers from two major problems when the operator has not been monitoring the team prior to the request for help. The first is its stationary nature: a limited, static view of the world results in ambiguity about the spatial relationships between objects and robots, especially in a high clutter environment with limited camera angles. The canonical solution is to shift a robot's viewpoint; however, if the robot is near obstacles which the human is unable to accurately localize, such uninformed motion could prove dangerous. A static view of the world also hampers the acquisition of situational awareness. As discussed in [2], time and the perception of temporal dynamics greatly influences the acquisition of Level 2 and 3 situational awareness.

A second limitation of a "traditional" teleoperation interface in this scenario is its lack of history and its lack of support for determining the intentions of the autonomous agents. One aspect of attaining situational awareness is determining what action the robot(s) were attempting to perform when they asked the human for help. This is often not obvious from a static view of the world: for instance, if a manipulator is in contact with an object, is it having trouble picking it up, or accurately placing it?

One approach to easing the acquisition of situational awareness is to maintain a buffer of information, which can be played back to the human when their assistance is required. This allows the human to "get up to speed" by allowing them to view information about the system as it approached the state that triggered the request for help. This makes it possible to both infer the robot(s) intentions and avoid needless motion through potentially hazardous terrain. The two obvious questions for implementing such a buffer are (1) what data is most relevant to attaining situational awareness, and (2) how much data it should buffer.

One may be tempted to include all available data in the interface on the theory that more data results in greater information intake by the human. However, the human brain's ability to winnow information from useless data is distinctly finite, and the operator will quickly become overwhelmed; Endsley terms this problem the "information gap" in [2]. Not only are the human's resources finite, but the ability of the robots to store and transmit large amounts of data is also restricted, especially when the robots are not colocated with the human. Bandwidth is always limited, and serves as a firm upper cap on the amount of data which may be presented to the human.

Alternatively, the human could be given their choice of display elements. However, since the human cannot know which are the most useful data streams *a priori* and the system cannot know which streams the human will choose, performance would suffer greatly. The human would often not have the proper data available, and the system would be unable to perform effective caching due to bandwidth limits, resulting in large delays before the human would be able to intervene.

The question of how much data to buffer is also more complex than it appears. While it may seem obvious that a longer buffer will result in a greater degree of situational awareness, and thus a faster response time, this is not necessarily the case. We hypothesize that there is a point at which longer playbacks provide no more useful information about the problem at hand. In addition, when considering efficiency, one must take into account not only how long it takes the human to react after viewing the playback, but also how long they spend watching the playback. There may be a tradeoff between playback time and time spent thinking about the situation, and it is possible that the optimal overall reaction time is not necessarily the case resulting in the minimum cognition time.

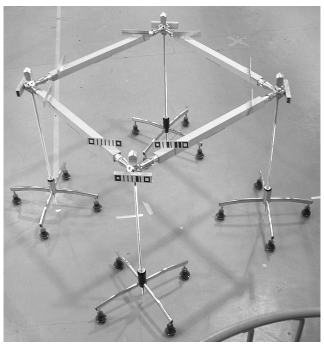
Although the specific answers to the questions of data relevance and buffer length are in part task- and domain-dependent, there are some principles that apply to a range of similar domains. In order to investigate these principles, we evaluate them using our own construction domain and robot team, and discuss our specific results and how they may apply to other similar scenarios.

4. EXPERIMENTAL DESIGN AND METHODOLOGY

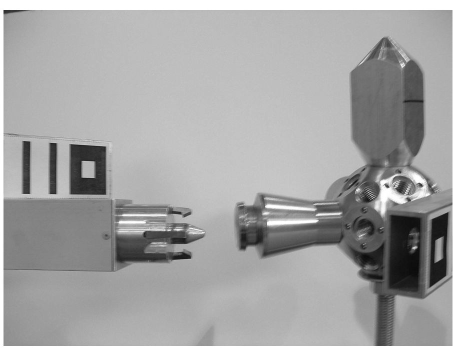
We conducted an experiment to assess how much and what types of information shown to a human operator correlate with how quickly the operator is able to gain situational awareness. In the experiment, the subjects were asked to observe a set of prerecorded data streams and then determine both why the autonomous system requested assistance and identify an appropriate action. Between trials, we vary which data streams are available to the subject, as well as the streams' length. When not responding to a simulated request for help, the subject performs a concentration-intensive distractor task, to simulate a multitasking operator.

4.1 Scenario and Robots

The assembly scenario used in this experiment involves four beams and four planarly compliant nodes that are assembled into a square structure (Figure 1(a)). In order to weakly simulate conditions in space, the nodes are supported by casters that roll easily along the floor. Thus, bracing of the nodes is required before the end of a beam may be inserted into the node (Figure 1(b)).



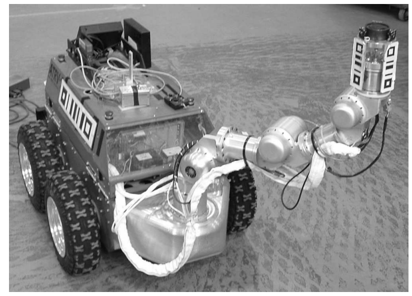
(a) The completed structure.



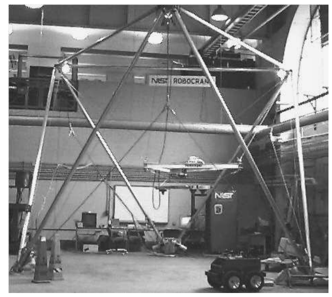
(b) Docking close-up

Figure 1: The fully assembled four-beam structure and a beam being inserted into a node.

We have decomposed this scenario into tasks that can be completed by agents fulfilling three different roles: an agent that provides information about the state of the world (the Roving Eye; Figure 2(c)), an agent that braces the nodes during docking (the Crane; Figure 2(b)), and an agent that does the actual manipulation and insertion of the beams into the nodes (the Mobile Manipulator; Figure 2(a)). Neither the Crane nor the Mobile Manipulator possess any extrinsic sensors that are available to the autonomous system, and must rely on positional data transmitted to them by the Roving Eye, which is equipped with stereo cameras.



(a) Mobile Manipulator



(b) Crane

(c) Roving Eye

Figure 2: The three robots used to build our square structure.

4.2 Interface

The information streams available to the human include three video feeds and one synthesized "technical drawing"-style visualizer (Figure 3). The video feeds are from one of the Roving Eye's cameras, a fisheye camera placed in the Crane looking down onto the workspace, and an external camera placed outside the workspace looking towards the structure. The Roving Eye's stereo pair is also used to estimate the relative positions of the various objects in its field of view [5]. This position information is in turn used to display the relative positions of the beam and node from above and in front of the beam in the visualizer ((4) in Figure 3). While somewhat noisy, this provides data that is at times not otherwise available to the user, due to the lack of depth perception from single cameras. Neither the camera on the Crane nor the external camera are utilized by the autonomous system.

4.3 Experimental Design

Our two experimental variables are the composition and length of the data feed that is presented to the subject. We chose to investigate four lengths (0, 5, 10, and 20 seconds) and four different combinations of available data (see Figure 3):

1. Roving Eye video only (RE)
2. Roving Eye video and the visualizer (RE/viz)
3. Roving Eye, Crane overhead, and external videos (RE/vid)
4. Roving Eye, Crane overhead, and external videos, as well as the visualizer (RE/viz/vid)

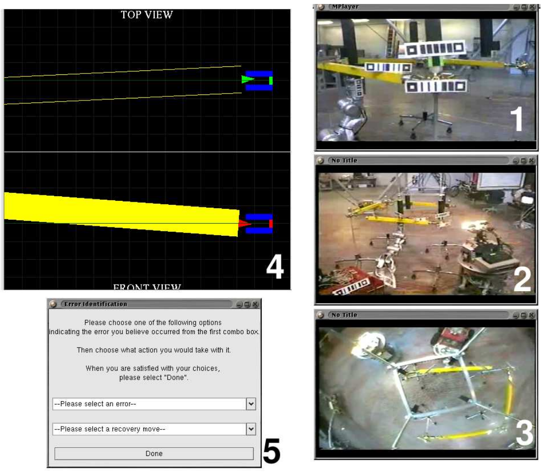


Figure 3: The subject interface, including three video streams (the Roving Eye's cameras (1), an external camera (2), and a Crane-mounted camera (3)), a synthesized view of the beam and node (4), and the error categorization / recovery action input dialog (5).

This yields a total of 16 different test conditions. The data feed combinations were chosen to allow the comparison of minimal data (1, above) against maximal data (4, above), as well as several points in between. They are then evaluated in conjunction with playback time of varying lengths, ranging from static feedback (the 0 second playback condition) to 20 seconds of feed.

The example requests for assistance used in the experiment were drawn exclusively from the task of docking one end of a beam with a node. This is a precise manipulation task performed by the Mobile Manipulator, and is rich in potential errors. This provided a large variety of situations in which the robots could request help, lowering the probability of the subject randomly guessing the correct answer. In addition, this placed all the examples within a constrained subset of the domain, which made training subjects much more tractable.

During each trial, the subject was asked to identify why the robot requested help during the observed docking and choose an appropriate low-level recovery action. The errors fell into four broad categories: false alarms, obscurements, interference by non-target objects, and interference by the target node. False alarms occurred when a successful docking had been incorrectly labeled an error by the system. An obscurement error occurred when the Roving Eye lost sight of the beam for any reason. Interference by non-target objects could have consisted of the beam hitting either another (non-target) node or the Crane's end effector. Finally, the beam could have become stuck in an undocked position on the target node, as a result of an error in the Roving Eye's data or the Mobile Manipulator approaching the node along an erroneous vector.

4.4 Experimental Procedure

The experimental procedure was a combination of training and testing. The subject's training began with reading a written overview of the task and hardware at hand, with the experimentor answering any questions.[1] The subject was then shown one example of each of the seven types of errors via the graphical interface (Figure 3), using the maximal data and 20-second playback condition. The experimentor discussed each example with the subject in order to ensure the subject understood each error's characteristics. These examples were not used during the test phase. If the subject did not feel fully trained by this point, the same training examples were repeated until the subject felt comfortable with the task.

After training (which typically took 20-30 minutes), the following test procedure was used. The subject first played Crack Attack, a Tetris-like game requiring significant concentration [8], for a time chosen from a normal distribution centered at one minute, with a standard deviation of 5 seconds. After this time had elapsed, the subject's display was switched to the interface (Figure 3), with the current condition's data streams visible . The length of playback associated with the current condition was immediately shown, with all displayed data streams synchronized. Once playback was complete, the classification entry dialog was displayed (note that this prevented the subject from choosing an error prior to the completion of playback). As soon as the subject selected one of the seven error classifications and one of the 13 recovery actions, they were returned to the distractor task. The entry dialog forced the user to first select an error, then a recovery action.

Each user was tested on four of the 16 conditions, with six examples of errors chosen per condition. Each set of six examples was chosen randomly without replacement from a pool of 29. In order to ensure that no one error type occurred more than once per test condition, when an example was picked and removed from the pool, all other examples of the same type were marked. Marked examples were removed from the pool until the six examples for the current test condition were selected, after which they were unmarked and returned to the pool. The entire experiment, including training and testing phases, consumed an average of 1.5 hours per subject.

To account for ordering effects, we applied Latin squares to both effects and ran the combined conditions. A Latin square is a statistical technique which allows experimentors to test effects while controlling for two other known sources of variation (here, intersubject variability and ordering effects). Since each subject was evaluated under four test conditions, 16 subjects were required to cover all the possible orderings. We evaluated 32 subjects in all. Our subjects were self-selected from the general Carnegie Mellon student body. None had prior experience with the task, and their backgrounds spanned the Carnegie Mellon population.

5. RESULTS

During the experiments, we recorded the time it took subjects to classify each example, both including and not including the time it took them to watch the data feed. We define "response time" as the time between when the data stream finished and when the users classified the current error via the dialog box. We now analyze this data in the context of the 16 test conditions (available data feeds and length of playback) using a univariate ANOVA test.

The results show that the data stream condition with the shortest response time was the one that displayed only the Roving Eye video

[1]Since a significant portion of the training consists of interactions between the experimentor and the subject, a single experimentor conducted all the experiments, in order to avoid training bias.

Table 1: Subject response time by data stream, without playback time (seconds)

	μ	σ
Roving Eye video	19.2	17.7
Roving Eye video / other video	26.9	30.0
Roving Eye video / visualizer	20.4	16.5
Roving Eye video / other video / visualizer	27.2	23.3

Table 2: Significance values between data stream conditions, without playback time

	RE	RE / vid	RE / viz	RE / viz / vid
RE	–	.000	.576	.000
RE / vid	.000	–	.003	.909
RE / viz	.576	.003	–	.000
RE / viz / vid	.000	.909	.002	–

(Table 1). This was very closely followed by the Roving Eye video and visualizer condition, while both conditions that included the remaining two video streams took significantly longer. The differences between the two sets of conditions were all statistically significant, using an LSD post hoc test (Table 2). When playback time was considered along with response time, the same significance held (Table 3, Table 4).

The results also confirmed our hypothesis that a longer video playback time results in faster user response time (Table 5). This trend was true for each condition; however, using an LSD post hoc test, significance was not found between the 10 and 20 second conditions (Table 6). When playback time was considered as part of the error response time, however, the resulting trends changed. Here, the fastest conditions were the 5 and 10 second playback conditions, with the 0 and 20 second playbacks taking significantly longer (Table 7, Table 8).

The ANOVA test also revealed trends in the interaction effects between the two experimental variables, with a significance of .083, both with and without playback time added in with the error response time. Figures 4 and 5 illustrate these effects. The graph of response time alone (Figure 4) suggests that the RE video / other video and the RE video / other video / visualizer conditions are more affected by video playback time than the other data stream conditions. It also shows that the data feed conditions have much less of an effect on users' mean response time when the playback video length is 20 seconds than when it is only a still frame. The graph that incorporates playback time (Figure 5) confirms these effects. These interactions will be discussed further in the next section.

Table 3: Subject response time by data stream, with playback time (seconds)

	μ	σ
Roving Eye video	28.6	17.6
Roving Eye video / other video	36.4	28.8
Roving Eye video / visualizer	29.8	16.3
Roving Eye video / other video / visualizer	36.6	23.3

Table 4: Significance values between data stream conditions, with playback time

	RE	RE / vid	RE / viz	RE / viz / vid
RE	–	.000	.576	.000
RE / vid	.000	–	.003	.914
RE / viz	.576	.003	–	.002
RE / viz / vid	.000	.914	.002	–

Table 5: Subject response time by data feed length, without playback time (seconds)

	μ	σ
still frame	35.0	26.5
5 seconds	23.3	19.6
10 seconds	18.7	21.3
20 seconds	16.7	18.5

Table 6: Significance values between data feed length conditions, without playback time

	still frame	5 seconds	10 seconds	20 seconds
still frame	–	.000	.000	.000
5 seconds	.000	–	.035	.002
10 seconds	.000	.035	–	.358
20 seconds	.000	.002	.358	–

Table 7: Subject response time by data feed length, with playback time (seconds)

	μ	σ
still frame	37.4	26.5
5 seconds	28.4	19.6
10 seconds	28.8	21.3
20 seconds	36.8	18.5

Table 8: Significance values between data feed length conditions, with playback time

	still frame	5 seconds	10 seconds	20 seconds
still frame	–	.000	.000	.801
5 seconds	.000	–	.850	.000
10 seconds	.000	.850	–	.000
20 seconds	.801	.000	.000	–

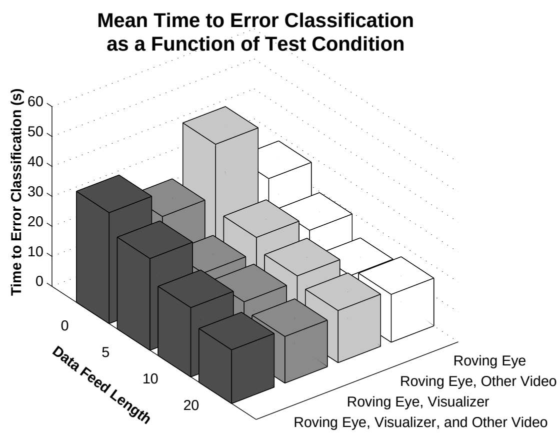


Figure 4: Average subject response time, by test condition

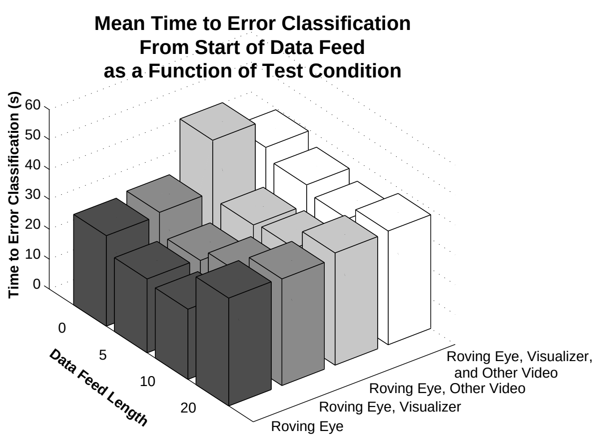


Figure 5: Sum of average subject response time and playback time, by test condition

6. DISCUSSION

In the terms of [6], the measured response times are an implicit measurement of situational awareness, as opposed to subjective (self-rating) and explicit (questionnaires administered during a suspension of the task) measurements. Most established methods for subjectively or explicitly measuring situational awareness, such as SART [9] and SAGAT [3], are designed to measure ongoing situational awareness during a long-term task, and are thus not immediately applicable to the domain we are investigating. However, we believe our implicit measurements to be a good measure of the ease or difficulty with which the subject attains situational awareness. By examining the time taken to classify the current error, we are able to directly measure how long it takes to attain Level 2 (comprehension) situational awareness, using Endsley's model [2]. Since the subject has been performing a high-concentration distractor task for at least a minute prior to each trial, we are confident that we are not missing any of the time the subject is using to attain situational awareness. In addition, because the subject's sole task once a trial begins is to determine the error, we also believe that we are not overestimating the time taken, except by the relatively constant factor of the time taken to select the chosen error from the interface. Thus, we believe that we are in fact able to capture accurately the time taken to achieve Level 2 situational awareness in our domain.

Looking only at the Roving Eye video and Roving Eye video / other video conditions, the results suggest that the simpler display (here, just the Roving Eye video) leads to the shortest response time. We believe that since subjects have limited visual information to consider while making their choice, it takes them less time to decide how to respond given the available information. When presented with more information, however, subjects took significantly longer to respond, suggesting that the extra videos add a significant processing overhead.

However, when the visualizer information is added to each of these conditions, user response time does not significantly increase. One possible explanation is that information presented by the visualizer is easier to process than information that must be extracted from the videos. This is especially true since the visualizer presents 3D information in a natural way, requiring little extra mental processing, whereas subjects are required to fuse the multiple video streams in order to extract 3D information from them. Another possible explanation is that subjects made little use of the visualizer, and instead concentrated their attention on the videos. In order to directly measure this, attention tracking data (such as that from a gaze tracking system) is needed. However, we may be able make this determination based on the subjects' error classification accuracy; presumably, subjects who utilized the information would have a lower error rate than those who did not.

We also conjecture that although the response time for the Roving Eye video is shorter than the other data stream conditions, it also leads to less accurate error classification. Further analysis of the data, including accounting for error classification accuracy, will yield statistics supporting or refuting this claim prior to the final submission of this paper.

While considering the results from the data feed length conditions, ignoring playback time, we can at first glance see that the longer the length of the buffer users are presented with, the better - there is a clear inverse relationship between the playback time and the time taken to select an error condition. An important aspect of these results to consider is that users' response times do not decrease linearly with respect to playback time; as the playback time increases, users' response time improves less and less. This suggests that at some point, user response time will plateau, making additional playback time ineffective. By adding the playback time to the error response time, however, we see that after a certain point, it is no longer optimal to increase the playback time. This is because it takes longer to both watch the display and respond than it does to watch a shorter display buffer and take a bit longer to respond. Thus, the data suggests that a playback in the range of 5 to 10 seconds is the most desirable option for this configuration. Although both conditions are roughly equal using just this data, further analysis of the data will more than likely reveal that accuracy improves with the longer playback clips until it plateaus at a certain point, making the 10 second playback the more desirable option. The additional dimension of accuracy allows explicit, principled tradeoffs between speed and accuracy to be made when designing teleoperation systems.

The interaction effects between the two experimental variables provided further insight into this situation. The increased sensitivity of the two data feed conditions that include all three video playbacks to data feed length supports the earlier conjecture that processing information in the form of raw video output takes longer than processing information from a simpler component, such as the visualizer. Also supported is our earlier theory about the human user's performance plateauing after viewing a certain length of playback. Because the difference between data stream conditions decreases as the playback time increases, it can be seen that users are becoming saturated with information, and that more information will most likely not decrease their response time any further.

Based on this data, we recommend that interfaces of systems such as ours include the equivalent of only one video display of the robot workspace. Supplemental information (depth perception, etc.), if needed, can then be provided by more processed, easier to understand displays, such as our visualizer. We can also recommend that each scenario be tested to determine how much playback time is required in the buffer in order to minimize the time necessary for the operator to gain situational awareness.

7. FUTURE WORK AND CONCLUSIONS

We were unable to log one obvious aspect of how situational awareness is attained: which data streams the subject was actually attending to at any given moment. This means that we are unable to distinguish between whether the subject was attending to a video that was of little utility for the current error or the subject was attending to a useful video, but did not comprehend the error. One approach that would allow us to separate these factors is to add a gaze tracker to the system.

As mentioned above, the response time of users under increasing data stream lengths probably plateaus after a certain point. It would be interesting to study further what exactly determines where this point is, and whether it is affected by other factors, such as the selected data streams and the speed of the robots in autonomous mode. It may be that a human assisting a robot team which moves quickly requires less absolute time to achieve situtional awareness. This would be the case if the important factor in the data streams' effect on situational awareness attainment is the observation of specific events, rather than merely observing for a given length of time. If this is the case, a slow-moving system can potentially improve the human's response time by playing back the buffer at a faster rate.

In this paper we have discussed the importance of quickly gaining situational awareness in systems involving sliding autonomy. In order for a human operator to be an effective team member in a system asking for help, they must be able to switch tasks, gain situational awareness, and diagnose the error as quickly as possible. We introduced various factors that affect a human operator's ability to quickly gain situational awareness of their workspace when shown a brief, partial history of the robots' movements. We varied which data feeds were shown and how long the feeds were, in an attempt to determine the optimal interface to our scenario and team. Human subject experiments have shown that an interface with one raw video and a visualization tool, as well as a 5 or 10 second playback of the buffer, is the best combination of components for users of our system to quickly gain situational awareness.

8. REFERENCES

[1] M. R. Endsley. A methodology for the objective measurement of situation awareness. *Situational Awareness in Aerospace Operations (AGARD-CP-478)*, pages 1/1 – 1/9, 1989.

[2] M. R. Endsley. *Situation Awareness Analysis and Measurement*, chapter Theoretical Underpinnings of Situation Awareness: A Critical Review. Lawrence Erlbaum Associates, Mahwah, NJ, 2000.

[3] M. R. Endsley, S. J. Selcon, T. D. Hardiman, and D. G. Croft. A comparative analysis of sagat and sart for evaluations of situation awareness. In *42nd annual meeting of the Human Factors and Ergonomics Society*, Chicago, October 1998.

[4] F. Heger, L. Hiatt, B. Sellner, R. Simmons, and S. Singh. Results in sliding autonomy for multi-robot spatial assembly. In *8th International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, Munich, Germany, September 5-8, 2005.

[5] D. Hershberger, R. Burridge, D. Kortenkamp, and R. Simmons. Distributed visual servoing with a roving eye. In *Proceedings of the Conference on Intelligent Robots and Systems (IROS)*, Takamatsu Japan, October 2000.

[6] A. T. Hjelmfelt and M. A. Pokrant. Coherent tactical picture. Technical Report CNA RM 97-129, Center for Naval Analyses, Alexandria, Virginia, 1998.

[7] J. W. Lasswell and C. D. Wickens. The effects of display location and dimensionality on taxi-way navigation. Technical report, Aviation Research Laboratory, Institute of Aviation, University of Illinois at Urbana-Champaign, May 1995. Prepared for NASA.

[8] D. R. Nelson and A. Sayman. Crack attack! manual. Game's home page: http://www.nongnu.org/crack-attack/, http://aluminumangel.org.

[9] R. M. Taylor. Situational awareness rating technique (sart): The development of a tool for aircrew systems design. *Situational Awareness in Aerospace Operations (AGARD-CP-478)*, pages pp. 3/1 – 3/17, 1990.

[10] E. K. Vogel, G. G. Woodman, and S. J. Luck. Storage of features, conjunctions, and objects in visual working memory. *Journal of Experimental Psychology: Human Perception and Performance*, 27(1):92–114, 2001.

[11] W. Wang and P. Milgram. Viewpoint optimization for virtual environment navigation using dynamic tethering - a study of tether rigidity. In *47th Annual Meeting of the Human Factors and Ergonomics Society*, Denver, Colorado, 2003.

[12] H. A. Yanco and J. Drury. "where am i?" acquiring situation awareness using a remote robot platform. In *IEEE Conference on Systems, Man and Cybernetics*, October 2004.