

Sliding Autonomy for UAV Path Planning: Adding New Dimensions to Autonomy Management

Lanny Lin, Michael A. Goodrich, Spencer Clark
Computer Science Department, Brigham Young University

Increased use of autonomy also increases human-autonomy interaction and the need for humans to manage autonomy. We propose a new variation of the concept of sliding autonomy that is useful for planning problems over a spatial region. In this sliding autonomy approach, the user can influence the behavior of the autonomous system via three categories of input: information, spatial constraints, and temporal constraints. We present a set of user interface designs to implement sliding autonomy for UAV (Unmanned Aerial Vehicle) path planning to support Wilderness Search and Rescue (WiSAR). Interactivities along these new dimensions allow the user to allocate degrees of authority and flexibility to the robot's algorithms. We analyze how this approach fits in the integration challenge guidelines we identified in our prior work and evaluate the usefulness of the approach against manual and simple pattern path planning methods with a user study. Results show that the sliding autonomy approach performs significantly better than the other two methods without increasing the users' mental workload, and the performance of the human-autonomy team outperforms either human or autonomy working alone. We also discuss some interesting observations from the user study.

Keywords: human-robot interaction, unmanned aerial vehicles, path planning, navigation, adjustable autonomy, supervisory control, sliding autonomy

1. Introduction

With the rapid advancement in technology, people are seeing increased use of autonomy to augment human abilities and support human decision-making in many application domains (e.g., (Chun, 2010; Casper & Murphy, 2003; Lin, Roscheck, Goodrich, & Morse, 2010; Robins, Dautenhahn, & Dickerson, 2009)). At the same time, increased use of autonomy also means increased human-autonomy interaction and increased need for humans to manage autonomy (Linbridge, 1983). Even for so-called fully autonomous systems, human input can potentially improve the system's performance and safety. The humans in such interactions manage autonomy, because "only people are held responsible for consequences (that is, only people can act as problem holders) and only people decide on how authority is delegated to automata" (Woods & Hollnagel, 2006; Bradshaw, Hoffman, Johnson, & Woods, 2013).

When humans manage autonomous systems, their managerial responsibilities often include

Authors retain copyright and grant the Journal of Human-Robot Interaction right of first publication with the work simultaneously licensed under a Creative Commons Attribution License that allows others to share the work with an acknowledgement of the work's authorship and initial publication in this journal.

monitoring the safety of the autonomous system, supervising autonomy to achieve acceptable performance, and making sure autonomy is working toward the collective goal of the overall system. In many emerging domains, the human operators are domain experts who can use domain-specific knowledge to assist the autonomous system when it deals with changing environments, uncertainty, and case-specific scenarios. Therefore, it is necessary to design tools and interfaces that enable human users to manage the autonomous behaviors of the system efficiently and effectively; such tools can improve task performance and the experience of the human operator in human-autonomy interaction. Wilderness Search and Rescue (WiSAR) is one such domain that could benefit from autonomy management tools when a mini-UAV (Unmanned Aerial Vehicle) is used in search.

Camera-equipped mini-UAVs can be useful tools in WiSAR operations by providing aerial imagery of a search area with the benefits of quick coverage of large areas, access to hard-to-reach areas, and lower cost than manned aircraft (Murphy et al., 2008; Goodrich et al., 2008). In fact Canadian mounties claim that they have successfully saved a person with a police drone in a recent rescue mission¹. UAV path planning is an important task because a good flight path can increase the probability of finding a missing person by making efficient use of the limited flying time. Various algorithms have been developed to support UAV path planning autonomy (e.g., (Bourgault, Furukawa, & Durrant-Whyte, 2003; Lin & Goodrich, 2009, 2014)). But how can a searcher incorporate domain expertise to manage UAV path planning autonomy efficiently and effectively without knowing how autonomy works behind the scene?

We propose a new autonomy management approach where the user can influence the behavior of an autonomous system along three new dimensions: 1) **Information Representation**: information used by the robot is presented to the human in a human-readable form, and the human directly modifies this information to effect change in robot behavior; 2) **Spatial Constraints**: a human can add constraints or priorities to different spatial regions, thereby affecting how the robot plans and performs its task; and 3) **Temporal Constraints**: A human can impose time limits for a subtask or impose ordering constraints on a subtask. We refer to this approach as *Sliding Autonomy* because, properly designed, it can allocate degrees of authority and flexibility to the robot's algorithms by adding or removing constraints, or by shaping input information. Indeed, we will explicitly use a **slider** as one GUI tool for managing UAV path planning.

As the human modifies information, adds priorities, or changes constraints, the sliding autonomy tool shows immediately how those changes influence the UAV's plan. This instant feedback provides the searcher the ability to perform "what-if" analysis and see the causal effect between his/her action and changes in autonomous behavior. This allows an interactive approach where autonomous algorithms perform tasks that they are good at and humans do tasks that they are good at, but in a collaborative and interactive way that avoids the pitfalls of simple task allocation (Sheridan, 1992). Properly done, the human-robot team should perform better than a human or robot working alone.

Many approaches to autonomy management already exist and are called many different things, such as *supervisory control* (Sheridan, 1992), *mixed-initiative* (Hearst, 1999), *collaborative control* (Fong, Thorpe, & Baur, 1999), *adjustable autonomy* (Dorais, Bonasso, Kortenkamp, Pelland, & Schrechenghost, 1998; Dorais & Kortenkamp, 2001) (also referred to as *sliding autonomy* (Dias et al., 2008) or *adaptive automation* (Rouse, 1988; Kaber, Riley, Tan, & Endsley, 2001)). The approach we propose falls under the category of *adjustable autonomy*. The three dimensions we identified are in addition to dimensions of *adjustable autonomy* identified by Bradshaw et al. so we design tools and algorithms that operate in a particular place in Bradshaw's taxonomy (Bradshaw et al., 2004).

In our previous work (Lin et al., 2010) we identified key elements of autonomy integration

¹<http://www.theverge.com/2013/5/10/4318770/canada-dragonflyer-drone-claims-first-life-saved-search-rescue>

challenges along two dimensions: *attributes of an intelligent system* (capability, information management, performance evaluation) and *organizational scale* (individual versus group), which can serve as guidelines in designing autonomous components and autonomy management tools. In this paper we extend the guidelines to include attributes needed when human and autonomy work collaboratively and analyze how our proposed sliding autonomy approach fits in the guidelines. By applying sliding autonomy to the UAV path planning task, we argue that this approach:

- enables the domain expert user to incorporate information only available to or understandable by the expert;
- is easy to understand without knowing how autonomy works behind the scene;
- lets the human do what the human is good at (planning strategically and balancing performance tradeoffs) and autonomy do what autonomy is good at (planning tactically), resulting in better performance than human-autonomy working alone;
- enables the user to achieve task goal (find the path that maximizes probability collected along the path) with system goal (finding the missing information quickly) when the user has more information or more up-to-date information than autonomy;
- and improves human's experience during the human-autonomy interaction.

To evaluate the usefulness of the proposed approach, we performed a user study and compared the sliding autonomy method against two other planning methods (manual and simple pattern path planning) in two WiSAR scenarios (a synthetic scenario and a real scenario). We measured each user's performance with each method and also the user's performance on a secondary task (answer questions in a group chat window). Experiment results show that the sliding autonomy method performed significantly better than the manual or simple pattern planning methods with no increased mental workload. The human-autonomy team also performed better than the human or autonomy working alone.

In Section 2 we explain how the proposed approach fits into the extended autonomy design guidelines and describe how a user can manage autonomy along each of the three new dimensions in the context of UAV path planning. Section 3 covers related work in literature. Section 4 lists our hypotheses followed by user study design in Section 5. Then we present experiment results in Section 6 and discuss our observations in Section 7. In Section 8 we conclude the paper and list possible future work.

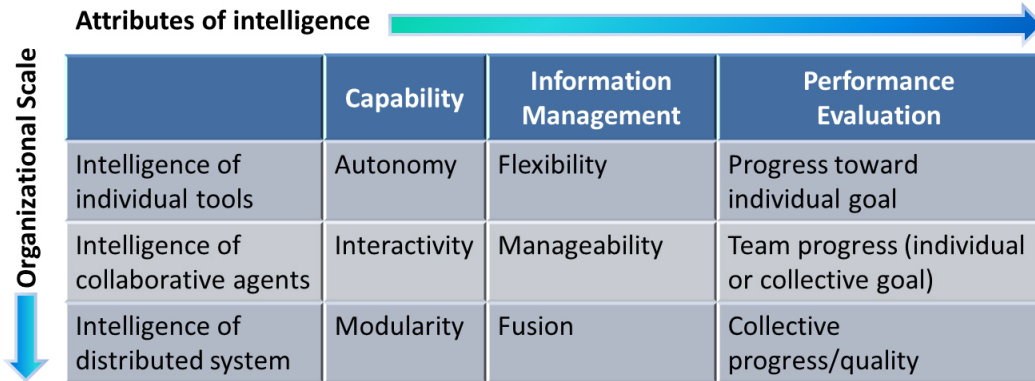
2. Autonomy Design Guidelines and New Dimensions

2.1 Autonomy Design Guidelines

In our previous work (Lin et al., 2010) we organized the challenges of autonomy and management tool design along two dimensions: *attributes of an intelligent system* (capability, information management, performance evaluation) and *organizational scale* (individual versus group), which can serve as guidelines in designing autonomous components and autonomy management tools. Here we extend this table by adding a row in the middle describing what attributes are needed when multiple agents work collaboratively (see Figure 1). A human-autonomy team working on the same task falls within this category.

As an individual tool, an autonomous component needs to be able to perform a task (**Autonomy**); the operator can match capability to task according to the information available to the operator, which requires that autonomy can be interrupted, paused, aborted, and resumed (**Flexibility**); the performance is evaluated to match individual task goal.

When a human-autonomy team works on the same task collaboratively, the autonomous component needs to provide interfaces so the human can interactively influence the autonomous behavior with human inputs (**Interactivity**); the human should be able to manage how autonomy works in



Organizational Scale ↓	Attributes of intelligence →			
	Capability	Information Management	Performance Evaluation	
	Intelligence of individual tools	Autonomy	Flexibility	Progress toward individual goal
	Intelligence of collaborative agents	Interactivity	Manageability	Team progress (individual or collective goal)
	Intelligence of distributed system	Modularity	Fusion	Collective progress/quality

Figure 1. Autonomy integration challenges defined along two dimensions. Horizontal dimension: attributes of intelligence. Vertical dimension: organizational scale.

order to jointly find a solution by utilizing **Information** only available to the human and/or feed information to autonomy in a representation that autonomy can understand (**Manageability**); and when performance is evaluated, the human operator can judge whether the individual goal aligns with the collective goal of the system.

As part of a larger distributed system, each autonomous component needs to be modular (**Modularity**), so they can be mixed and matched to support different user roles; information from various sources need to be combined and presented to one or multiple users in a **Fusion**; and performance of the system needs to be evaluated as a whole.

This paper focuses on the middle row of the guidelines: intelligence of collaborative agents (human-autonomy team). The three dimensions we propose are ways path planning autonomy can be managed, and our path planner interface is designed to accept human inputs along the three dimensions to provide interactivity. The human can also incorporate information from various sources and influence the behavior of path planning autonomy by allocating degrees of authority and flexibility, making sure the task goal aligns with the ultimate goal of finding the missing person quickly.

2.2 Information Representation Dimension

Many path planning algorithms use a probability distribution map that shows where are likely places to find the missing person. We also designed our planning algorithms to support a task difficulty map: a spatial representation (one minus sensor detection probability in different parts of the search region. For example, probability near the last known position (LKP) of the missing person is normally high (low task difficulty); and the probability of detecting the missing person in a dense vegetation area from camera footprints is usually low (high task difficulty). The objective of path planning is to find a path that maximize the detection probability of the missing person given a fixed flying time.

These two maps are in human-readable form and are used directly by path planning autonomy. They can be systematically generated based on terrain features and vegetation data (Lin & Goodrich, 2010, 2014). However, the searcher likely wants to include his/her domain expertise (past experience, knowledge of the search region, etc.) and additional information (maybe new evidence found during the search) in the planning. These types of information are not directly understandable by autonomous algorithms, but the searcher can incorporate them into the probability distribution map

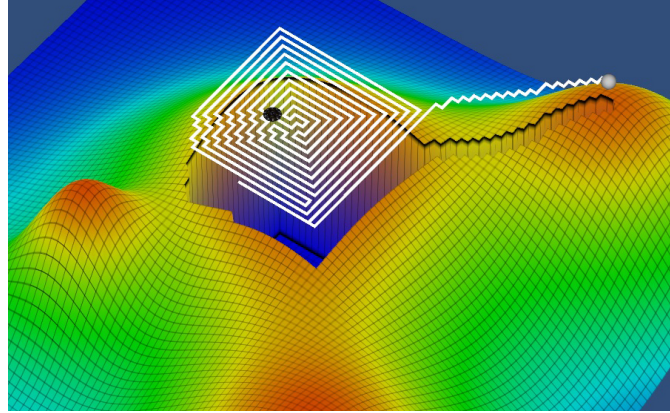


Figure 2. A screen capture of the sliding autonomy tool showing a 20-minute path segment. The 3D surface shows the probability distribution map. The UAV icon in the middle indicates the start point of the path segment and the sphere on the right indicates the end point.

and the task-difficulty map using map editing tools², this influence the behavior of path planning autonomy.

Marking an area with high probability, the searcher indirectly tells the UAV to treat the area with high priority; marking an area with high task-difficulty, the UAV might make multiple passes over the area to search more thoroughly. The 3D surface in Figure 2 shows an example probability distribution map where hills (red) indicate high probability and the flat area (blue) indicates low probability.

By managing information representation, the searcher can quickly figure out how his actions will affect the behavior of autonomy, even though he/she has no idea about how autonomy works behind the scene. Our interface supports autonomy management along this dimension. But we choose to not include it in the current user study (due to the one hour limit per test subject) and leave the evaluation in a separate user study.

2.3 Spatial Constraints Dimension

The searcher can also influence the behavior of path planning autonomy by setting/changing spatial constraints. Spatial constraints can be in the forms of start/end points of the path segment or task-specific zones.

Setting an end point in an area is a way for the searcher to indirectly tell path planning autonomy that the area should have higher priority than other areas. For example, if a piece of clothing is found by the ground team, the searcher can force path planning autonomy to go visit that area by setting an end point there. Because the UAV must allocate part of the fix-length flight time to reach this specified area, some areas that had good trade-offs before become costly and are no longer attractive to path planning autonomy. This way the searcher can also direct path planning autonomy to avoid certain areas (maybe the area has been thoroughly searched by the ground team). In Figure 2, the UAV icon in the middle indicates the start point of the path segment and the sphere on the right side indicates the desired end point.

The end point can be dragged around the search region and path planning autonomy suggests different paths accordingly. This capability enables the user to adjust how much freedom is granted

²Such as these tools at <http://tech.lannyland.com/demos.html>.

to autonomy. When the end point is close to the start point, autonomy has greater authority and flexibility in creating paths. If the end point is far from the start point, authority and flexibility for autonomy is reduced because a major part of path planning is simply moving the UAV toward the end point with the shortest path.

A task-specific zone can be a no-fly zone, a coverage zone, or a sampling zone. A no-fly zone is a pretty straight-forward way to restrict the UAV from visiting certain areas. The decision might be for safety reasons or part of the searcher's strategic planning depending on resource allocation. A coverage zone requires the UAV to fully cover the area, a sampling zone only asks the UAV to collect a few sensor samples from the zone, so the visit can be very brief. A task-specific zone can be dragged around and the searcher can also change the shape and/or size of the zone to influence the path generated by autonomy.

The interactive ability to move the end point around (or changing the shape of the no-fly zone) and see immediately how the change would affect the UAV path recommended by path planning autonomy gives the user the power to perform "what-if" analysis. It also allows the user to see the causal effect between his/her action and changes in autonomous behavior.

Spatial constraints are easy to understand, so the searcher knows how these constraints will affect the behavior of path planning autonomy. By managing autonomy along this dimension, the searcher has another way to incorporate additional information to the path planning task, improve task performance, and align the task goal with the overall goal of finding the missing person quickly.

In our user study we fixed the start point of the path to the center of the map because that was the last known position of the missing person. The searcher can set the end point for the current path segment anywhere on the map, and this end point automatically becomes the start point for the next path segment. This enabled the searcher to move an end point once a path segment was planned to reduce computation, but lets the user reset the end point and re-plan. Task-specific zone was not included in this user study (it was tested in a separate user study (Clark & Goodrich, 2013)).

2.4 Temporal Constraints Dimension

In the UAV path planning problem, temporal constraints can be time limit for a subtask (path segment), subtask ordering, or valid time window.




With the time limit constraint, the searcher can decide how much flight time to allocate to a path segment out of the total flight time. This enables the searcher to break the path planning task into multiple subtasks and then plan each path segment separately. In our interface design we let the searcher control time allocation to autonomy using a slider, and as the searcher moves the slider, path planning autonomy shows how suggested path segment changes respectively. Similar to the spatial constraints, this instant feedback enables "what-if" analysis and provides instant feedback on the causal effect between searcher action and changes in autonomous behavior.

For example, for a 60-minute total flight with an end point set to the probability hill on the right (Figure 2), the searcher can move the slider to set time limits and see immediately what path segment autonomy would suggest. The path segment shown is when 20 minutes are allocated. If the searcher is happy with the suggestion, he approves the path segment. The UAV moves to the end point in the path planner and "vacuums up" the probability along the path (how much can be vacuumed up is determined by the task-difficulty map). Then the searcher works with autonomy to plan path for the remaining 40 minutes. The two (or more) path segments are joined to form the final path.

Subtask ordering constraint adds temporal dependency to the subtasks (e.g., the sampling task must be flown before the coverage task). It lets the searcher directly specify priorities in different search areas. Valid time window constraint specifies at what time interval a subtask can be completed, so it is more restrictive than subtask ordering constraint.

By managing autonomy along the temporal constraint dimension, the searcher can break the

path planning task into subtasks and incorporate additional information into the path planning task. The user study described in this paper includes the time limit constraint. Subtask ordering and valid time window constraints are evaluated in a separate user study (Clark & Goodrich, 2013)).

In the example shown in Figure 2, the combination of the information representation, end point constraint (spatial), and time limit constraint (temporal) allows the searcher to cover the middle area pretty well and  move to the search area on the right ~~ready~~ to search there. If no end point constraint is set, autonomy might decide to search the probability hill on the left with a low time limit or  all at the bottom with a high time limit. If more flight time is allocated with the set end point, autonomy might start searching the area on the right. Less time allocation reduces the authority and flexibility of autonomy, and forces autonomy to focus more on the local area; more time allocation increase authority  flexibility, so autonomy has more freedom on deciding what areas to cover. Instant ~~feed back~~ on path changes when the search moves the endpoint or varies the time limit lets the searcher interactively review multiple options and select the path segment that fits best with his/her strategic planning. This design enables human to plan more strategically (prioritizing areas in the entire search region) while autonomy works more tactically (covering the current search area well), using strengths of each when they work collaboratively. Ideally such a human-autonomy team should work better than either human or autonomy working alone.

3. Related Work

Drucker defines automation as a “concept of the organization of work (Drucker, 2006).” Goodrich and Schultz define the HRI problem as “understanding and shaping the interactions between one or more humans and one or more robots” (Goodrich & Schultz, 2007). They also specified robot-assisted search and rescue as a key area for HRI research. In their 1978 seminal paper, Sheridan and Verplank propose the idea of a *level of autonomy* spectrum, with full teleoperation at one end and full autonomy at the other (Sheridan & Verplank, 1978). In the middle, the robot could suggest actions to humans or make decisions before informing humans. Parasuraman et al. extended this one-dimensional spectrum to four different broad functions: information acquisition, analysis, decision selection, and action implementation (Parasuraman, Sheridan, & Wickens, 2000). Sheridan proposes *supervisory control*, in which a human divides the task into a sequence of subtasks that the robot is capable of performing, and the human then provides guidance when the autonomous system cannot solve a problem on its own (Sheridan, 1992). In contrast to the top-down philosophy of supervisory control, a *mixed-initiative* approach advocates the idea of dynamically shifting tasks when necessary (Hearst, 1999). *Collaborative control*, which can be thought of as an instance of mixed-initiative interaction, is a robot-centric model; instead of the human always being in-charge, the robot is treated as a peer and can make requests to humans through dialogs (Fong et al., 1999). *Adjustable autonomy* (Dorais & Kortenkamp, 2001) (also referred to as *sliding autonomy* (Dias et al., 2008) or *adaptive automation* (Rouse, 1988)) is another type of mixed-initiative interaction, one that enables the human-automation team to dynamically and adaptively allocate functions and tasks among team members.

Dorais et al. discuss a framework for human-centered autonomous systems for a manned Mars mission (Dorais et al., 1998). The system enables users to interact with these systems at an appropriate level of control but minimize the necessity for such interaction. Bradshaw et al. discuss principles and pitfalls of adjustable autonomy and human-centered teamwork, and then present study results on so-called “work practice modeling” and human-agent collaboration in space applications (Bradshaw et al., 2003). Kaber et al. describe an experiment simulating an air traffic control task where manual control was compared to Adaptive Automation (AA) (Kaber, Wright, Prinzel III, & Clamann, 2005). Results suggest that humans perform better with AA applied to sensory and psychomotor information-processing functions than with AA applied to cognitive functions; these results also sug-

gest that AA is superior to completely manual control. Brookshire et al. present preliminary results for applying sliding autonomy to a team of robots performing coordinated assembling work to help the system recover from unexpected errors and to thereby increase system efficiency (Brookshire, Singh, & Simmons, 2004). Dias et al. identified six key capabilities that are essential for overcoming challenges in enabling sliding autonomy in peer-to-peer human-robot teams (Dias et al., 2008). Bradshaw et al. propose two dimensions of Adjustable Autonomy (descriptive and prescriptive) to address the two senses of autonomy (self-sufficiency and self-directedness) and discuss how permissions, obligations, possibilities, and capabilities can be adjusted (Bradshaw et al., 2004). Bradshaw et al. also summarized some widespread misconceptions on autonomy and listed seven deadly myths of “autonomous systems” (Bradshaw et al., 2013).

The human is an integral part of the human-autonomy team. When working with autonomy, the human often takes on the supervisor role. Bainbridge points out that automation requires the human operator to take additional management responsibilities (Bainbridge, 1983), and Sartar identified two automation management policies: *management by consent* and *management by exception*, defining whether the human always retain authority or can the system take initiative (Sarter, 1998). For complex automation, the human tends to rely on his/her *mental models* (Norman, 1983) to manage the system.

UAV technology has emerged as a promising tool in supporting WiSAR (Murphy et al., 2008; Bourgault et al., 2003). The goal of our research is to support fielded missions in the spirit of Murphy’s work (Casper & Murphy, 2003). Many path planning algorithms in the literature address obstacle avoidance while planning a path to reach a destination using A* (Quigley, Barber, Griffiths, & Goodrich, 2005), LRTA* (Howlett, McLain, & Goodrich, 2006), D* (Stentz, 1997), Voronoi diagrams (Bortoff, 2000; Beard et al., 2005), or probability roadmaps and rapidly-exploring random tree (RRTs) (Pettersson & Doherty, 2006). Bourgault et al. (Bourgault, Goktogan, Furukawa, & Durrant-Whyte, 2004; Bourgault, Furukawa, & Durrant-Whyte, 2006) describe how to use a Bayesian model to create paths for a single UAV or multiple coordinated UAVs to maximize the amount of probability accumulated by the UAV sensors. The algorithms we used in this paper are algorithms designed from our previous work (Lin & Goodrich, 2009, 2014) using techniques such as global warming technique, convolution, Gaussian mixture models, and Evolutionary Algorithm.

4. Hypotheses

A user study is performed to evaluate the usefulness of the sliding autonomy approach. More specifically we verify the following hypotheses:

H1: The sliding autonomy method performs better than either the manual path planning method or a semi-autonomous pattern path planning method in both low information and high information scenarios.

H2: The sliding autonomy method performs better than autonomy working alone in both low information and high information scenarios.

H3: The sliding autonomy method does not increase the mental workload of the operator when compared against manual and pattern methods.

5. User Study Design

We performed a 2×3 within-subject design with 2 scenarios (easy vs. difficult) and 3 planning methods (manual, pattern, and sliding autonomy). All participants completed all 6 exercises. The order of the scenarios and planning methods was counterbalanced to reduce learning effect. We recruited a total of 26 college students (14 males and 12 females) between the age of 19 and 30 (average 22.89).

After the demographic survey, each participant completed four 5-minute long non-able training sessions (one for each planning method with no task-difficulty map, and one for manual

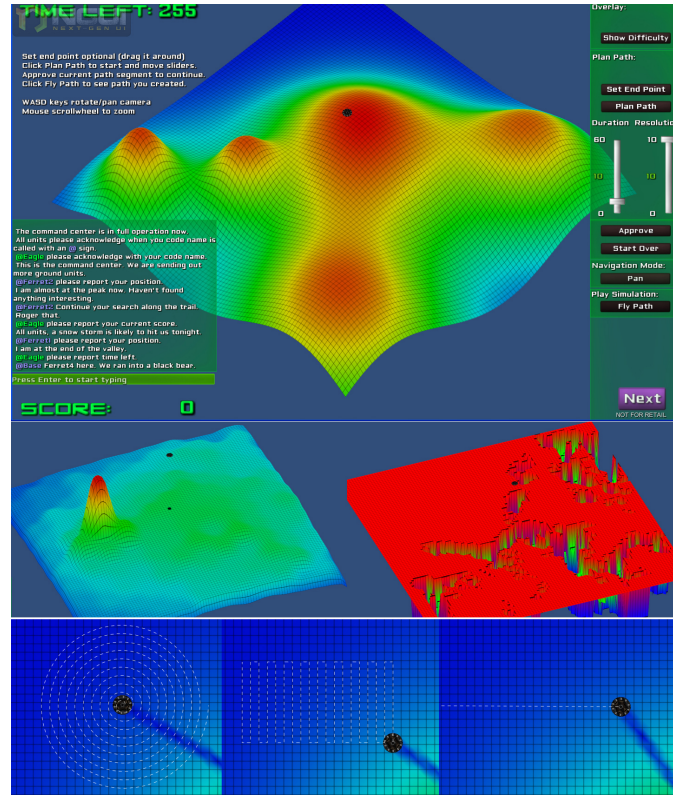


Figure 3. Top: User study simulation interface with the sliding autonomy method showing the probability distribution map for scenario 1. Middle left: Probability distribution map for scenario 2. Middle Right: Task-difficulty map for scenario 2. Bottom: The three patterns available to user with the pattern planning method, spiral, lawnmower, and line.

with a task-difficulty map) and then completes the 6 exercises. Each participant has up to 5 minutes for each exercise. Once the participant is happy with the path generated, he/she can finish the exercise early. We choose this design because we do not want the user to put all effort into completing the secondary task once he/she considers the primary task completed, which would skew the measurements on secondary task performance. At the end of each exercise, the participant completes a NASA TLX survey. Then at the very end of the user study, the participant fills out a survey describing his/her subjective preference with the three planning methods.

5.1 Simulation Environment

The user study is conducted in a 3D simulation environment (see Figure 3) where both the probability distribution map and the task-difficulty map are displayed as 3D surfaces with a color map (red means high and blue means low). The user can switch between the two maps at any time and rotate/pan/zoom a map at will. The UAV is a hexacopter capable of flying in all directions or hovering in the same spot.

With the **manual** planning method, the user flies the UAV with arrow keys in a sped up fashion. The user can switch between two flying modes (turn and strafe) and four camera views (global,

behind, bird's eye, and free form). The user can also pause/resume the flight for the secondary task or better planning.

With the **pattern** planning method, the user chooses from spiral, omniscient, and line patterns (see Figure 3 bottom) and joins these patterns to form the path. Cursor position marks the end of the path segment and determines the size of the pattern. The user can also undo the last path segment planned. This planning method is “semi-autonomous” because the patterns are generated automatically without manually setting waypoints.

With the **sliding autonomy** method (see Figure 3 top), the user can set an endpoint (optional), and then drag the left slider to change the amount of time allocated to autonomy. Path suggested by autonomy changes as the slider moves. The slider's max value always reflects the remaining flight time (in minute). If the user is happy with the current path segment, he/she approves it, the UAV moves to the end of the path segment, and the process repeats until all flight time has been planned. The algorithm used is the LHC-GW-CONV algorithm (Goodrich, 2009, 2014), because it is the fastest algorithm out of all the algorithms we designed.

With all three planning methods, the user can choose to start over at any time during the exercise, and can restart as many times as exercise time allows. We record the best path out of all tries.

5.2 Scenarios

The user study contains two WiSAR scenarios, a synthetic case (see Figure 3 top) with no task-difficulty map (assuming uniform detection probability), and a real WiSAR scenario (see Figure 3 middle) with a task-difficulty map, in which an elderly couple was reported missing near the Grayson Highlands State Park in Virginia (Koester, 2008)³.

Scenario 2 is clearly more complex than scenario 1 because the user also has to consider the different detection probability defined by the task-difficulty map. We refer to scenario 2 as the high information scenario and scenario 1 as the low information scenario.

5.3 Secondary Task

In each exercise, each participant also performs a secondary task, so we can measure the user's mental workload. In a group chat window (see Figure 3 top) when the user's code name is called, the user has to type answers to simple questions. Roughly every 3 seconds a message is sent to the chat window, and every 5th message asks the user a simple question (4 per minute). For the same scenario and the same planning method, all users received the same set of chat messages.


We choose to use a group chat window as the secondary task because this is typical in WiSAR operations. We also designed the chat messages to simulate a real WiSAR search. The user is asked to acknowledge connection and report path planning status periodically. This design ensures that the secondary task is ecologically valid (Vicente, 1997; Rasmussen, Pejtersen, & Goodstein, 1994).

5.4 Measures


We use the following five measurements for the primary path planning task:

- **Percent score:** In each exercise, an exercise score is computed by summing the amount of probability collected by the UAV if it followed the path planned. The user's best score for each exercise (out of multiple tries) is normalized by dividing the best score from all users for the same scenario to compute the percent score. This way we can compare planning methods across scenarios.

³The probability distribution map used for this scenario (Figure 3 middle left) was generated using a Bayesian model (Lin & Goodrich, 2010). The map has been evaluated at George Mason University's MapScore web portal (Cawi, Jones, & Twardy, 2012) and performed better than most other models evaluated, scoring 0.8184 on a [-1,1] scale where the higher the score the better. <http://sarbayes.org/projects/>. The task-difficulty map (Figure 3 middle right) was generated using vegetation density data downloaded from the USGS web site and categorized into three difficulty levels (sparse, medium, and dense, with detection probability of 100%, 66.67%, and 33.33% respectively).

- **Time spent:** How much time is spent with each exercise.
- **Try count:** How many times the user tried in each exercise. Note that because the manual planning method takes much longer to plan a path than the other two methods by design, this measurement is used mainly to compare between the pattern and sliding autonomy planning methods.
- **Mouse clicks per try:** How many times the user left-clicked the mouse within a try.  In this measurement is used to compare pattern and sliding autonomy planning methods because manual planning methods does not require a lot of mouse clicks by design.
- **NASA-TLX raw score:** The sum of user subjective evaluation of cognitive workload in six dimensions normalized to a 100-point scale.

The following two measurements are used for the secondary task:


 **Percent of questions missed:** What percentage of questions directed to the user were missed before user completed the exercise. Here we do not measure the percent of questions answered correctly because all the questions are very simple and all users answered the questions correctly.

• **Chat latency:** The number of seconds between the time a question was presented to the user and the time when the user answered the question.


6. Results and Analysis


We analyzed user study data with a mixed measures analysis of variance (ANOVA) and report results in this section.

6.1 Comparing Across Scenarios

~~During the user study, each participant worked through two WiSAR scenarios. In scenario 1 (low information) no task-difficulty map is used. In scenario 2 (high information) because partial detection is enforced, the amount of probability that can be collected (exercise score) in scenario 2 is much lower than in scenario 1. In order to show a fair comparison across scenarios, we use percent score instead, which compares the participant's score to the best performing participant's score in the same scenario.~~ 

No statistically significant differences were found across scenarios for percent score, time spent, and try count. Average percent score is very close in both scenarios (76.99% and 74.17%) because we normalize exercise scores against the highest user score for the same scenario. There is a slight difference of 26 seconds in average time spent, which is not significant compared to the 300 seconds given for each exercise. Average try count in both scenarios are also close enough (3.08 and 2.67).

Mouse clicks per try for the two scenarios are significantly different ($F[1, 25] = 28.65, p < .0001$) indicating scenario 2 required more physical work from each participant than scenario 1. This result matches the difficulty of the two scenarios where each participant created more path segments with the pattern and sliding autonomy planning methods in scenario 2 because it uses a complex task-difficulty map. 

NASA TLX scores are also significantly different ($F[1, 25] = 31.35, p < .0001$) between the two scenarios where the average score difference is 9.98 (out of a total of 100 points), almost a full "pip" on the TLX survey, indicating that on average each participant felt his/her cognitive workload was much higher in the high information scenario. 

The percent of questions missed is almost identical between scenarios (54.88% and 54.90%), and the chat latency is also very close (10.39 and 11.17 seconds). This shows that participants on average performed about the same with the secondary task across scenarios.

~~If we look at the performance of only the sliding autonomy planning method across scenarios, results follow the same trend.~~

Move this sentence to here

I'm trying to simplify this section and make

Table 1: Comparing across planning methods (SE stands for standard error)

	M	P	SA	SE	Significance
% Score	59.40	72.75	94.60	1.39	$F[2, 50] = 223.03, p < .0001$
Time spent	243.35	240.02	228.37	12.06	$F[2, 50] = 1.16, p = .32$
Try count	1.75	3.56	3.31	0.43	$F[2, 50] = 9.47, p = .0003$
Clicks/try	13.01	35.64	25.58	2.90	$F[2, 50] = 19.47, p < .0001$
NASA TLX	61.51	49.18	48.86	2.81	$F[2, 50] = 14.15, p < .0001$
% Q. missed	52.94	56.69	55.04	5.17	$F[2, 50] = 1.26, p = .29$
Chat latency	10.39	11.17	10.92	0.65	$F[2, 50] = 0.46, p = .63$

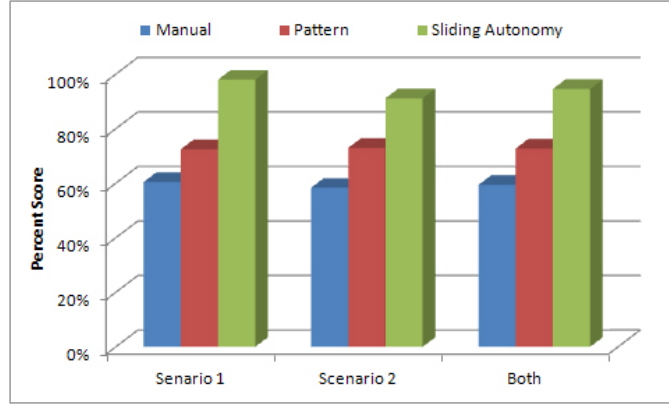


Figure 4. Performance differences for the three path planning methods.

6.2 Comparing Across Planning Methods

For each scenario, three path planning methods were used (manual, pattern, and sliding autonomy). Table 1 lists comparison among these three methods.

Percent score differences are statistically significant ($F[2, 50] = 223.03, p < .0001$) with sliding autonomy (94.60%) performing better than pattern (72.75%) and manual (59.40%). As shown in Figure 4, this trend is also clear in both scenario 1 and 2 individually. Therefore, user study results show strong support for our first hypothesis: sliding autonomy method performs better than either the manual method or the pattern method in both low information and high information scenarios.

Statistically significant differences were also found in mouse clicks per try ($F[2, 50] = 19.47, p < .0001$). The manual method uses arrow keys to fly the UAV around and only uses mouse clicks when switching camera modes or stop the timer in order to perform the secondary task. By design, this method does not use a lot of mouse clicks. Pattern and sliding autonomy methods both use mouse clicks for the actual path planning task, and the pattern method clearly generated more mouse clicks per try (35.64) than the sliding autonomy method (25.58). Two factors might have contributed to this difference, first one being that the pattern method allows undoes while the sliding autonomy method does not support this function. Also with the sliding autonomy method a participant can drag the slider and see different suggestions from path planning autonomy, which does not require a lot of mouse clicks.

NASA TLX raw scores show significant differences ($F[2, 50] = 14.15, p < .0001$) among the

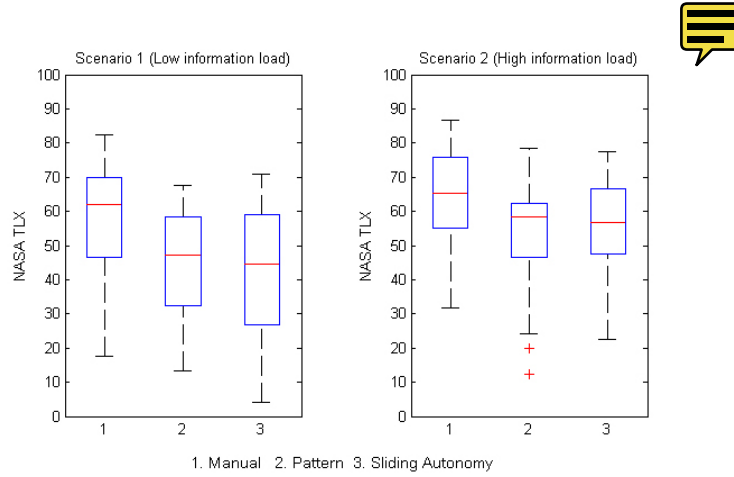


Figure 5. Box plots of the NASA TLX scores for each scenario.

three methods, with the manual method showing the highest cognitive mental workload (61.51), a full “pip” more than the other two methods on the TLX survey. However, the average score difference between the pattern method and the sliding autonomy method is small (49.18 vs. 48.86). Figure 5 shows the box plots of the NASA TLX scores for each scenario. Participants felt that the pattern method and the sliding autonomy method required about the same amount of cognitive mental workload in each scenario, and the manual method required a lot more mental workload in both scenarios.

For all three planning methods, participants performed about the same on the secondary task, as shown by percent of questions missed and chat latency in Table 1. Combining this with percent score and NASA TLX we can conclude that sliding autonomy performed best without increasing participants’ mental workload, which support our third hypothesis: Sliding autonomy method does not increase the mental workload of the operator when compared against manual and pattern methods.

If a participant allocates the entire flight time (60 minutes) to path planning autonomy (LHC-GW-CONV algorithm), the path is generated by full autonomy without any human input. The percent score for this path would be 96.13% for scenario 1 and 78.83% for scenario 2. This is better than average participant performance with the manual and pattern methods in both scenarios (see Figure 6). If we compare each participant’s percent score against full autonomy for all 26 participants, in scenario 1, none could outperform full autonomy using the manual or pattern methods, but 23 (88.46%) were able to perform better with sliding autonomy (human-autonomy team). In scenario 2, no one did better using the manual method, 5 (19.23%) did outperform full autonomy using the pattern method, and 24 (92.31%) were able to beat full autonomy using the sliding autonomy method. Table 2 summarizes the comparison. User study results clearly show that the human inputs made significant differences in path planning performance and human-autonomy team outperformed autonomy working alone. This also strongly supports our second hypothesis: Sliding autonomy method performs better than autonomy working alone in both low information and high information scenarios.

6.3 Additional Factors

We also performed ANOVA analysis on some additional factors that might create differences: gender, experience in video games, order of the scenarios, and whether participants used full autonomy

Table 2: Percent of participants outperforming autonomy with each method

	Manual	Pattern	Sliding Autonomy
Scenario 1 (Low)	0%	0%	88.46%
Scenario 2 (High)	0%	19.23%	92.31%

with the sliding autonomy method. No significant differences were found for these factors overall, across scenarios, or across methods. There is also no significant correlation (-0.23) between percent of questions missed in the secondary task and the NASA TLX raw scores.

7. Discussion

7.1 Planning methods Characteristics

Manual Method

With the manual method, the user uses arrow keys to move the UAV around to create a path, so by design the method is very intuitive, flexible, and requires more physical interactions (keyboard, not mouse clicks). But in order to plan a 60-minute path in 2 minutes, the UAV has to move very quickly in the simulation environment. We made sure each user can complete at least two tries using this method while performing the secondary task simultaneously. High UAV speed makes it harder to navigate accurately. Many participants called the arrow keys too “sensitive” and recommended slowing down the UAV.

Because of the time pressure, the path planning task is more of a continuous process. Therefore when errors are made, it is too costly to start over (we did not provide an undo function). Participants also felt that they do not have the luxury to pause the planning in order to think through strategies in the head—it has to be done while a user is moving the UAV around. Naturally, when this continuous process is frequently interrupted by the secondary task where a user has to pause planning and answer questions in the group chat window, user frustration goes high.

More physical work, higher frustration, and lower performance score are the main factors contributing to a much higher NASA TLX score for the manual method. Although this planning method supports four camera views and two flying modes, most participants chose to stick with free form view where the entire search region is displayed and with strafe mode to avoid getting confused with UAV orientation. During training, participants actually had one extra session with the manual method, but this method still performed the worst.

Pattern Method

With the pattern method, the user joins a mixture of three patterns (spiral, lawnmower, and line) together to form the final path. This is more of an episodic process, so it is very easy to pause in the middle of the planning and shift attention to the secondary task. There is also less time pressure because the user can quickly plan for the remaining time with just one big spiral (or lawnmower) in one click. Therefore, the user has plenty of time for many tries with different strategies.

In the post user study survey, many participants commented that with the spiral and lawnmower pattern it is really easy to run out of time. They suggested adding the ability to allocate time to the patterns similar to the sliding autonomy method. This means that with this method, a user enjoys the systematic coverage of an area but has a hard time estimating how much time it takes the UAV to cover the area following the pattern. Several participants also suggested adding more patterns to the method.



The pattern method is the only method ~~supporting undo~~. This ability ~~surely had impacts on~~ number of mouse clicks per try and participants' preference over the three planning methods. Another interesting observation is that participants seemed to be overly optimistic about their performance using the pattern method. For ~~example~~, although sliding autonomy performed better than pattern in all scenarios for all participants, ~~in 46.15% of the times, participants rated pattern as performing better than sliding autonomy in NASA TLX scores; and also in the post user study survey; 26.92% participants (not the majority, but over a quarter of the population) actually believed the pattern method creates best paths.~~

Sliding Autonomy Method

Similar to the pattern method, the sliding autonomy method is also episodic. The ~~stopping~~ in the middle of the planning to answer questions for the secondary task was easy. ~~And it only takes a few clicks to let autonomy plan path for the remaining time, so there is not much time pressure and the user can have many tries.~~

Because the user does not know how autonomy works behind the scene, many participants ~~were~~ surprised by the path recommended by autonomy, and feel that autonomy did not do what they ~~had~~ ~~tell~~ it to do. For example, when a user sets the end point in region A, autonomy might plan a path that spends most of time in a seemingly unrelated region B and only goes toward region A at the end of the path, because such a path is more efficient (scores higher). In such cases, the slider becomes the only tool that lets the user “force” autonomy to do what the user wants, and path planning turns into a fight between the human and autonomy. However, the instant feedback (displaying path and the predicted “vacuuming effect”) does help the user figure out why autonomy would suggest something different, and some participants were glad that autonomy suggested better paths they had not considered.

Most participants were generally happy with the path segment recommended by autonomy covering a local region, even when the region is in an irregular shape (not a circle or rectangle). Many participants also expressed that they did not have enough control over the path generation and recommended adding the ability to include constraints such as “middle points” where the path segment has to go through these ~~mid~~ ~~points~~. In fact, such “middle points” can already be achieved with the current method by setting ~~end~~ ~~points~~. Several participants complained that this method does not have the undo function. With both the pattern and sliding autonomy methods, many participants expressed the desire to be able to modify the path after it is generated.

User Preferences

In scenario 2 where a task-difficulty map ~~is used~~, ~~because the probability map is similar to a unimodal distribution (see Figure 3 middle left), most participants chose to first cover the probability hill using the probability distribution map, and then stuck to the task-difficulty map view to plan the remaining path. This behavior appeared in all three planning methods. If the probability distribution map is more complex with many probability hills, then likely the u~~ ~~ll be switching between the two maps back and forth in order to incorporate information. Some suggested showing both maps side by side or have a way to combine the two maps into one. These ideas are worth exploring in future user studies.~~

In the post user study survey, the majority of the participants think manual is the easiest to learn (53.85%), pattern is the easiest to use (57.69%), and sliding autonomy performed the best (65.38%). However, most participants preferred the pattern method (69.23%) out of all three. We believe the inability to undo, ~~not able to move the end~~ ~~once path is planned~~, and operator-induced oscillation when moving the slider ~~all had negative impacts on participants' preference over the sliding autonomy method.~~ ~~Once these functions are in place, the sliding autonomy method will~~

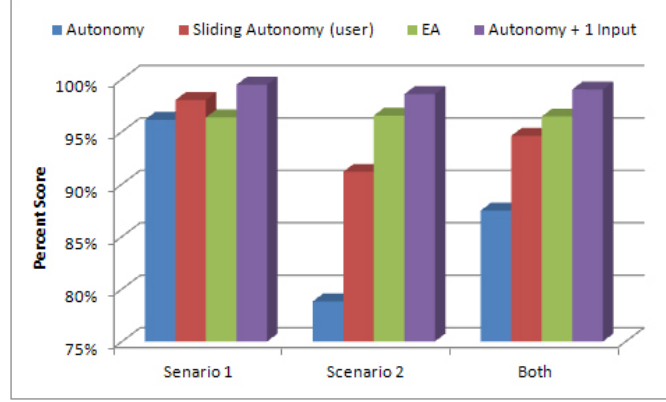


Figure 6. Comparing sliding autonomy performance against various markers.

Table 3: Percent of participants outperforming autonomy performance markers

	Autonomy	EA	Autonomy+1
Scenario 1 (Low)	88.46%	88.46%	7.69%
Scenario 2 (High)	92.31%	26.92%	15.38%

become more preferable than the pattern method.



Reliance on autonomy

The algorithm we used for the sliding autonomy method is the LHC-GW-CONV algorithm. We selected this algorithm because of its fast speed. If speed is not a concern, then the Evolution Algorithm (EA) we developed (Lin & Goodrich, 2009) can autonomously generate even better paths. For example, the percent score for EA would be 96.36% for scenario 1 and 96.54% for scenario 2. In scenario 1, the score is only slightly better than full autonomy in sliding autonomy (96.13%); but in scenario 2, the score is much better than full autonomy (78.83%). We also used the sliding autonomy method and generated paths for both scenario 1 and 2 with only one human input, an end point. Using the best score out of 3 tries (roughly equal to the average number of tries in the user study), we also computed the percent score for this autonomy+1 human input approach: 99.47% for scenario 1 and 98.58% for scenario 2. Using the EA and Autonomy+1 scores as additional markers, we plotted participants average performance in each scenario against the markers. Figure 6 shows the result.

First, sliding autonomy (human-autonomy team) outperformed autonomy (LHC-GW-CONV algorithm) in both scenarios. Sliding autonomy also outperformed EA in scenario 1 (low information). In scenario 2, the performance of sliding autonomy is not very far from EA (5.36%), and the difference is even smaller (1.85%) when averaged over both scenarios. However, the most interesting observation is that autonomy+1 actually outperformed all others in both scenarios (99.47% for scenario 1 and 98.58% for scenario 2). Although a few participants did score higher than autonomy+1, the difference is less than 1.5%. Table 3 lists what percentage of participants outperformed full autonomy, EA, and autonomy+1.

What Figure 6 suggests is that with the sliding autonomy method, it does not need a lot of

humans to perform very well. Instead of spending the effort creating many path segments and set many end points, it is probably more rewarding to search for the right region to set just one or two constraints. However, 88.68% of the participants gave more than 1 input when they used sliding autonomy (81.13% for 2 inputs and 69.81% for 3 inputs). In the post user study survey, on 46.15% of the participants acknowledged using full autonomy with the sliding autonomy method. These numbers indicate the human's under-reliance (Bradshaw et al., 2013) on autonomy. When using the sliding autonomy method, a good strategy is actually to start with autonomy (as the worst scenario) and then see how additional human inputs can improve the path.

Lee and See propose that because people respond to technology socially, trust guides reliance when unanticipated situations make it impractical or impossible to understand automation (Lee & See, 2004), and defined *distrust*, "trust falls short of the automation's capabilities", and *overtrust*, "trust exceeds system capabilities". Bradshaw et al. (Bradshaw et al., 2013) had similar definitions and used the term *under-reliance* for distrust, instead. Both suggested that trust in automation should be "calibrated". Hoffman et al. (Hoffman, Johnson, Bradshaw, & Underbrink, 2013) suggest "active exploration for trusting" (AET) and hope this approach can promote both trust "calibration" and appropriate reliance. We believe that the sliding autonomy approach we propose can be useful in this aspect, because as the user is moving the slider, he/she is calibrating his/her reliance on path planning autonomy. In our user study, all participants only had 30 minutes of training before using the sliding autonomy method. We speculate that as the user gets more familiar with the sliding autonomy approach in the long run, he/she would be able to calibrate reliance better. However, validation of this claim is beyond the scope of this paper and will be part of a future long-term user study.

When more complex probability distribution maps and task-difficulty maps are used, it is possible that the human user will see more value in the sliding autonomy approach because combining this much information in human mind and then decide tradeoffs in path planning required much more computation. And when the human user finds himself/herself incapable of such tasks, he/she might rely on autonomy to help solve the puzzle, and the sliding autonomy approach becomes more useful in calibrating reliance. Validating this hypothesis is another natural extension of the present work.

7.3 Why human-autonomy team performs better?

User study results show that the human-autonomy team outperformed either human or autonomy working along. But were they able to achieve this? We believe that the reason is: through the interactive interface, the sliding autonomy approach enabled the human to focus on what human is good at and autonomy to focus on what autonomy is good at. Bradshaw et al. point out (Bradshaw et al., 2013): "Humans, though fallible, are functionally rich in reasoning strategies: their powers of observation, learning, and sensitivity to context." Our observation suggests that human is really good on two things: planning strategically and recognize bad path segments.

The sliding autonomy method lets the user plan at a higher abstract level by specifying priorities in search sub-regions and how well each sub-region should be covered. Then how to cover the sub-region well given a fixed flight time is left for autonomy to handle. Autonomy, on the other hand can generate a path that covers a sub-region (or some nearby sub-regions) precisely and quickly, and can handle all kinds of irregular sub-region shapes. Therefore, the sliding autonomy method combines the strengths of both human and autonomy.

A human user is also very good at recognizing bad moves in solutions suggested by autonomy. The sliding autonomy approach enables the human user to select from a bunch of suggested paths. Selecting a path segment with fewer bad moves (which might not actually be bad in autonomy's perspective with just the current path segment) will probably increase the chance of a good final

path. So again, the sliding approach takes advantage of the strengths of both human and autonomy.

7.4 Why similar secondary task performance in all three methods?

The pattern and sliding autonomy methods are episodic, suggesting that it is easier for the user to pause planning and shift attention to the secondary task of answering questions in the group chat window. However, user study data show that there is no significant differences in secondary task performance across all three path planning methods.

The manual method requires a lot of continuous keyboard interaction (great physical demand and temporal demand) to move the UAV around. However, it does not actually require much mental demand and effort because the planning process is more sporadic and spontaneous. If a mistake is made, because there is no way to correct it, the user quickly stops worrying about it and moves on. The low mental demand and effort make monitoring the group chat window an easy task, even though switching back and forth between primary task and secondary task is very frustrating.

With the pattern and sliding autonomy methods, path planning is more like piecing together a puzzle. The user is deeply drawn into problem solving, constantly comparing tradeoffs, which actually requires more mental involvement. With the sliding autonomy method, the user is interacting with complicated algorithms, so while planning a path, the user is also trying to build a mental model of how autonomy works. As a result, the user actually paid less attention to the secondary task. Fighting with autonomy when human and autonomy had disagreements also drew user attention away from the group chat window. But when the group chat window catches the user's attention, he/she can perform the secondary task leisurely.

David Woods and Eric Hollnagel describe the law of stretched systems (Woods & Hollnagel, 2006): "every system is stretched to operate at its capacity; as soon as there is some improvement, for example in the form of new technology, it will be exploited to achieve a new intensity and tempo of activity." With the pattern and sliding autonomy methods, users had more tries and evaluated more options and tradeoffs. With the sliding autonomy method, users played more with spatial and temporal constraints, and evaluated more paths suggested by path planning autonomy, which resulted in better quality paths at the cost of no performance increase in the secondary task.

8. Conclusions and Future Work

In this paper we propose a new autonomy management approach, sliding autonomy, which lets the user influence the behavior of the autonomous system along three new dimensions: information representation, spatial constraints, and temporal constraints. We extend the autonomy design guidelines in our prior work by adding a new row for intelligence of collaborative agents (human-autonomy team), and explain how the three new dimensions fit into the guidelines when we apply the proposed approach to the task of UAV (Unmanned Aerial Vehicle) path planning to support Wilderness Search and Rescue (WiSAR). We present interface designs that let the user allocate degrees of authority and flexibility to the robot's algorithm through interactivities along these new dimensions. Experiment results from a user study validate our hypotheses and show that the sliding autonomy method performs significantly better than either the manual or pattern path planning method without increasing the user's mental workload, the human has better interaction experience, and human-autonomy team outperforms either human or autonomy working alone.

Possible future work include evaluating the sliding autonomy approach in a long-term study and see how the user's reliance can be calibrated. More complicated WiSAR scenarios can be tested and see how that affects the human-autonomy interaction and the performance of the human-autonomy team. Another user study can evaluate how well the human-autonomy team works with outdated information representation using the sliding autonomy approach. It would also be interesting to

investigate the human-autonomy interaction experience when the ability to modify information representation is added to the tool.

Acknowledgments

This work was partially supported by National Science Foundation under grant number 0534736 and _____, and under grant number _____ from the Army Research Laboratory. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsoring organizations.

References

- Bainbridge, L. (1983). Ironies of automation. *Automatica*, 19(6), 775–780.
- Beard, R., Kingston, D., Quigley, M., Snyder, D., Christiansen, R., Johnson, W., ... Goodrich, M. (2005). Autonomous vehicle technologies for small fixed-wing UAVs. *AIAA Journal of Aerospace Computing, Information, and Communication*, 2(1), 92–108.
- Bortoff, S. (2000). Path planning for UAVs. In *Proceedings of the American Control Conference* (Vol. 1, pp. 364–368).
- Bourgault, F., Furukawa, T., & Durrant-Whyte, H. (2003). Coordinated decentralized search for a lost target in a Bayesian world. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- Bourgault, F., Furukawa, T., & Durrant-Whyte, H. (2006). Optimal search for a lost target in a Bayesian world. In *Field and Service Robotics* (pp. 209–222).
- Bourgault, F., Goktogan, A., Furukawa, T., & Durrant-Whyte, H. (2004). Coordinated search for a lost target in a Bayesian world. *Advanced Robotics*, 18(10), 979–1000.
- Bradshaw, J., Feltovich, P., Jung, H., Kulkarni, S., Taysom, W., & Uszok, A. (2004). Dimensions of adjustable autonomy and mixed-initiative interaction. *Agents and Computational Autonomy*, 235–268.
- Bradshaw, J., Hoffman, R., Johnson, M., & Woods, D. (2013). The seven deadly myths of autonomous systems. *Intelligent Systems, IEEE*, 28(3), 54–61.
- Bradshaw, J., Sierhuis, M., Acquisti, A., Feltovich, P., Hoffman, R., Jeffers, R., ... Van Hoof, R. (2003). Adjustable autonomy and human-agent teamwork in practice: An interim report on space applications. *Agent Autonomy*, 243–280.
- Brookshire, J., Singh, S., & Simmons, R. (2004, March). Preliminary results in sliding autonomy for coordinated teams. In *Proceedings of The 2004 Spring Symposium Series*.
- Casper, J., & Murphy, R. (2003). Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 33(3), 367–385.
- Cawi, E., Jones, N., & Twardy, C. R. (2012). MapScore: Probability map evaluation for search & rescue. In *Virginia Search & Rescue Conference*. Conference Presentation presented at the Virginia Search & Rescue Conference, Holiday Lake, VA.
- Chun, A. (2010). Optimizing limousine service with AI. In *Proceedings of the Innovative Applications of Artificial Intelligence Conference*.
- Clark, S., & Goodrich, M. (2013). A hierarchical flight planner for sensor-driven UAV missions. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication* (pp. 509–514).
- Dias, M., Kannan, B., Browning, B., Jones, E., Argall, B., Dias, M., ... Stentz, A. (2008). Sliding autonomy for peer-to-peer human-robot teams. In *Proceedings of the Intelligent Conference on Intelligent Autonomous Systems* (p. 332).
- Dorais, G., Bonasso, R., Kortenkamp, D., Pelland, B., & Schrechenghost, D. (1998). Adjustable autonomy for human-centered autonomous systems on Mars. In *The First International Conference of the Mars Society*.
- Dorais, G., & Kortenkamp, D. (2001). Designing human-centered autonomous agents. In *Advances in Artificial Intelligence. PRICAI 2000 Workshop Reader* (pp. 321–324).
- Drucker, P. (2006). *The practice of management*. Harper Paperbacks, NYC, NY.

- Fong, T., Thorpe, C., & Baur, C. (1999). Collaborative control: A robot-centric model for vehicle teleoperation. In *AAAI Spring Symposium: Agents with Adjustable Autonomy*.
- Goodrich, M., Morse, B., Gerhardt, D., Cooper, J., Quigley, M., Adams, J., & Humphrey, C. (2008). Supporting wilderness search and rescue using a camera-equipped miniUAV. *Journal of Field Robotics*, 25(1-2), 89-110.
- Goodrich, M., & Schultz, A. (2007). Human-robot interaction: A survey. *Foundations and Trends in Human-Computer Interaction*, 1(3), 203-275.
- Hearst, M. (1999). Mixed-initiative interaction. *Intelligent Systems, IEEE*, 14(5), 14-23.
- Hoffman, R., Johnson, M., Bradshaw, J., & Underbrink, A. (2013). Trust in automation. *Intelligent Systems, IEEE*, 28(1), 84-88.
- Howlett, J., McLain, T., & Goodrich, M. (2006). Learning Real-Time A* path planner for unmanned air vehicle target sensing. *Journal of Aerospace Computing, Information, and Communication*, 3(3), 108-122.
- Kaber, D., Riley, J., Tan, K., & Endsley, M. (2001). On the design of adaptive automation for complex systems. *International Journal of Cognitive Ergonomics*, 5(1), 37-57.
- Kaber, D., Wright, M., Prinzel III, L., & Clamann, M. (2005). Adaptive automation of human-machine system information-processing functions. *Human Factors*, 47(4), 730.
- Koester, R. (2008). *Lost person behavior: A search and rescue guide on where to look - for land, air and water*. dbS Productions LLC, Charlottesville, VA.
- Lee, J., & See, K. (2004). Trust in automation: Designing for appropriate reliance. *Human Factors*, 46(1), 50.
- Lin, L., & Goodrich, M. (2009). UAV intelligent path planning for wilderness search and rescue. In *Proceedings on the IEEE/RSJ International Conference on Intelligent Robots and Systems* (p. 709-714).
- Lin, L., & Goodrich, M. (2010). A Bayesian approach to modeling lost person behaviors based on terrain features in wilderness search and rescue. *Computational and Mathematical Organization Theory*, 16(3), 300-323.
- Lin, L., & Goodrich, M. (2014). Hierarchical heuristic search using a Gaussian Mixture Model for UAV coverage planning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*. (Submitted, under 2nd round of review.)
- Lin, L., Roscheck, M., Goodrich, M., & Morse, B. (2010). Supporting wilderness search and rescue with integrated intelligence: Autonomy and information at the right time and the right place. In *Proceedings of the AAAI Conference on Artificial Intelligence, Special Track on Integrated Intelligence*.
- Murphy, R., Steimle, E., Griffin, C., Cullins, C., Hall, M., & Pratt, K. (2008). Cooperative use of unmanned sea surface and micro aerial vehicles at hurricane Wilma. *Journal of Field Robotics*, 25(3), 164-180.
- Norman, D. (1983). *Some observations on mental models*. Lawrence Erlbaum Associates, Mahwah, NJ.
- Parasuraman, R., Sheridan, T., & Wickens, C. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 30(3), 286-297.
- Pettersson, P., & Doherty, P. (2006). Probabilistic roadmap based path planning for an autonomous unmanned helicopter. *Journal of Intelligent and Fuzzy Systems*, 17(4), 395-405.
- Quigley, M., Barber, B., Griffiths, S., & Goodrich, M. (2005). Towards real-world searching with fixed-wing mini-UAVs. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- Rasmussen, J., Pejtersen, A., & Goodstein, L. (1994). *Cognitive systems engineering*. Wiley-Interscience, Hoboken, NJ.
- Robins, B., Dautenhahn, K., & Dickerson, P. (2009). From isolation to communication: A case study evaluation of robot assisted play for children with autism with a minimally expressive humanoid robot. In *Proceedings of the International Conferences on Advances in Computer-Human Interactions* (p. 205-211).
- Rouse, W. (1988). Adaptive aiding for human/computer control. *The Journal of the Human Factors and Ergonomics Society*, 30(4), 431-443.
- Sarter, N. (1998). Making coordination effortless and invisible: The exploration of automation management strategies and implementations. In *CBR Workshop on Human Interaction with Automated Systems*.
- Sheridan, T. (1992). *Telerobotics, automation, and human supervisory control*. The MIT Press, Cambridge, MA.

- Sheridan, T., & Verplank, W. (1978). *Human and computer control of undersea teleoperators* (Tech. Rep.). DTIC Document.
- Stentz, A. (1997). Optimal and efficient path planning for partially known environments. In *Intelligent unmanned ground vehicles* (pp. 203–220).
- Vicente, K. (1997). Should an interface always match the operators mental model? *CSERIAC Gateway*, 8(1), 1–5.
- Woods, D., & Hollnagel, E. (2006). *Joint cognitive systems: Patterns in cognitive systems engineering*. CRC Press, Boca Raton, FL.
-

Authors' names and contact information:

Lanny Lin, Brigham Young University, Provo, UT, USA. Email: lanny@lannyland.com.

Michael A. Goodrich, Brigham Young University, Provo, UT, USA. Email: mike@cs.byu.edu.

Spencer Clark, Brigham Young University, Provo, UT, USA. Email: spengy@gmail.com.