Analysis of a Gesture-based Interface for UAV Flight Path Generation

Meghan Chandarana[1], Erica L. Meszaros[2], Anna Trujillo[3], and B. Danette Allen[3]

***Abstract*— Traditionally, flight paths for unmanned aerial vehicles (UAVs) are generated offline by trained pilots and engineers using assumed environmental conditions, terrain and obstacles as constraints. As new applications for UAVs emerge, their user base shifts from one of operators with knowledge of low level systems to that of non-experts. These new operators require a more intuitive method for building desired UAV flight paths such that they can leverage the full capabilities of the vehicle without needing to understand its system complexities. We present a gesture-based natural language interface for defining trajectory segments using a library of twelve simple hand gestures. A user study is presented to analyze the effectiveness, ease-of-use and accuracy of the gesture-based interface as compared to a baseline mouse interface. We explore differences seen between subjects given their hand dominance, their prior UAV flight experience (or lack thereof), and whether they chose to sit or stand while using the gesture interface. Given limited training time, subjects were able to accurately define an average of 74.36% of trajectory segments. Overall the user study highlights the favorable potential for the use of the gesture-based interface as an alternative input modality, as well as, feedback for future interface improvements and training methods.**

I. INTRODUCTION

Continued advancement in radio and communication technology as well as controllers and interfaces has made equipment more affordable and easier to use then ever before. As a result, new applications for unmanned aerial vehicles (UAVs) are rapidly emerging in both the civilian and non-civilian sectors [1]. Traditionally, applications like search and rescue (SAR) [2], disaster relief [3], and intelligence, surveillance and reconnaissance (ISR) [4] missions are planned and executed by highly trained pilots and engineering specialists. Autonomous systems allow specialists to preprogram UAV coordination, flight paths, mission objectives and required parameters [5]. In SAR and ISR applications pilots and engineers develop intelligent strategies for searching predetermined areas of interest (AOI). These strategies are adapted for the number of UAVs set to be deployed. Each UAV utilizes on-board sensors and navigation systems to find and track a given target or location. Fig. 1 displays a sample mission AOI where three UAVs are tasked with searching for the source of a pollutant with a sweeping pattern (left) and replanning their trajectories to track the source once it it located (right) [6]. Sensor data is fused throughout the mission to improve efficiency.

[1]This author is with the Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA, USA. Email: mchandar@cmu.edu.
[2]This author is with the Department of Social Science, University of Chicago, Chicago, IL, USA. Email: elmeszaros@uchicago.edu.
[3]These authors are with National Aeronautics and Space Administration's Langley Research Center, Hampton, VA, USA. Email: {a.c.trujillo, danette.allen}@nasa.gov.

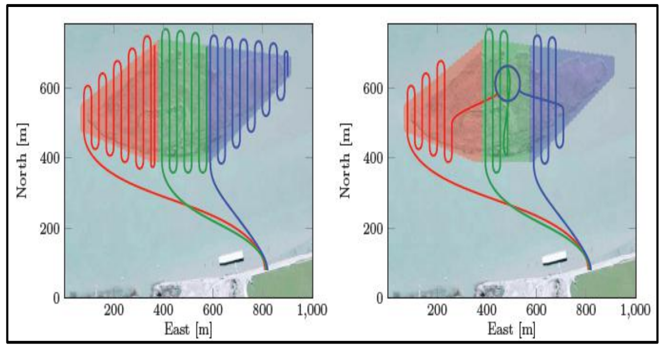


Fig. 1: Coordinated UAV search pattern for three vehicles within a predefined area of interest (AOI) [6].

More recently, applications like atmospheric data collection are expanding the core of the UAV user base from that solely trained specialists to include non-expert UAV users like scientists. Scientists look to leverage autonomous UAV technology to replace traditional data collection methods like air balloons, satellites and manned aircrafts, whose usual aim is to measure trends over time in a set of predefined AOIs (Fig. 1). These outdated technologies are costly, require an extended period of time to collect samples, and often only operate with a single sensor, thereby making correlative data collection laborious and troublesome [7]. The use of UAVs would give scientists a method for taking correlative data – required for more comprehensive studies – in situ using multiple vehicles. Additionally, real-time replanning allows for data-driven sampling.

With current interface and mission planning tools skilled pilots and engineers use their domain knowledge in UAV systems and guidance, nagivation and control to define end-to-end UAV missions. Researchers in the area of autonomous aerial mission planning utilize key insights and understanding of path planning schemes and vehicle performance (gained over years of experience). In most instances, scientists do not have the piloting and controls background required to understand the complex low-level commands needed to run UAV systems. Currently, manned science missions are planned and coordinated with a team of trained specialists. With scientists playing the role of mission manager, route planning of complex flight paths are negotiated within the team to achieve the desired goal of the mission while simultaneously maintaining safe and flyable trajectories (given the environment and known obstacles). In addition, missions are often generated and modified in extreme environments (e.g., cargo plane) where common interfaces like mouse and keyboard systems face significant

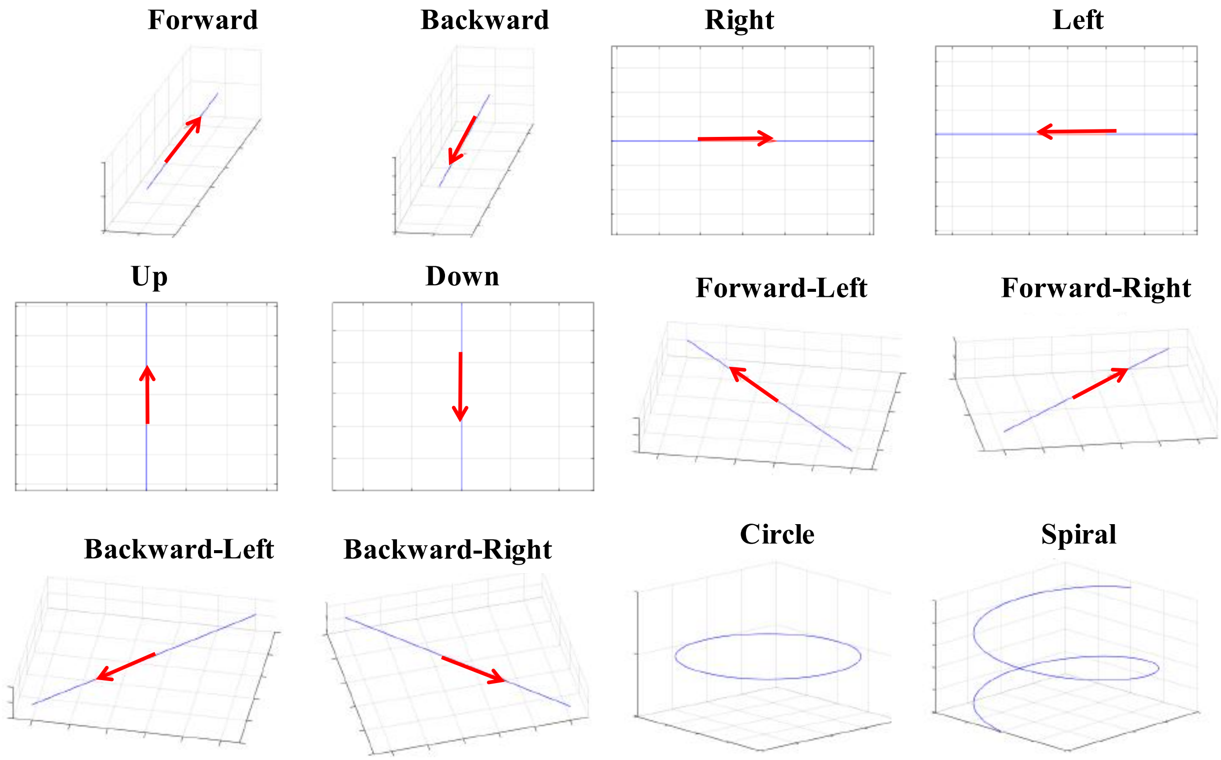


Fig. 2: Library of 12 trajectory segments developed in [5].

challenges like vibration. Therefore, to realize robust and easy-to-use systems that reduce the dependency on specialists, future interaction schemes must move away from traditional and arduous methods [2]. Specifically, they must provide more natural and intuitive interfaces for defining coordination schemes and mission objectives, as well as constructing desired flight paths. Interfaces that embrace natural and intuitive input modalities increase system efficiency and are more easily usable by a broader user base [8][9][10].

This work aims to explore how natural language can be used to develop a more intuitive interface for UAV mission management. We specifically examine the viability of a gesture-based interface in the context of UAV flight path generation. By leveraging gestures we can simulate common communication schemes seen in human-human interactions. The remainder of this paper evaluates the ease-of-use and efficacy of an adaptation of our previously developed gesture-based natural language interface [5] in comparison to a mouse-based baseline interface [11]. An initial analysis of the user study presented in this paper was explored in [11]. This paper provides a detailed analysis of the gesture user study by exploring how prior experience, hand dominance (i.e., right handed or left handed), and a user's choice to sit or stand while using the gesture-based interface affected performance and user workload.

The paper is organized as follows. Section II reviews current gesture-based UAV interfaces. Section III describes the proposed gesture-based interface and mouse-based baseline. Sections IV outlines the experimental setup. The results and discussion are presented in Sections V and VI. Lastly, Section VII provides a conclusion discusses future work.

II. Related Work

Over the years many "natural" interfaces have been developed, such as [12], where users controlled a graphical user interface (GUI) via a combined speech and gesture interface. Various interfaces allow users to directly define UAV flight paths, such as a speech-based interface [13] and a 3D spatial interface [14]. Although humans typically use both speech and gesture to communicate and convey ideas between each other, gestures are more widely used to represent complex ideas. Therefore, gesture-based interfaces have been increasingly implemented for human-robot interaction [15]. Previous interfaces analyze how various high level commands can communicate intent in human-robot teaming without specifically defining how the robot should move [16][17]. Initial human-UAV interaction research has explored interaction schemes in the context of a collocated UAV. Ng and Sharlin developed a socially motivated gesture-based interaction scheme for collocated UAVs based on a falconry metaphor [18]. Cauchard et al. show that humans naturally choose to interact with collocated drones as they would another person or pet [19]. The work of Suárez Fernández et al. seeks to provide UAV users a framework for flexible input modality/modalities for direct control of UAV movement depending on their application [20].

A variety of gesture-based human-robot input methods have been used in the past. These methods often restricted the natural arm or hand movement of the user by expecting them to wear or hold a sensor [21][22]. Gesture-based interfaces eventually implemented systems with unmounted sensors. Initially these systems relied on full body movements [23]. Some systems used static hand poses to program by demonstration [24][25] or encode complex, indirect movement [26]. None of the previous methods focused on using a simplistic, unmounted sensor to build complex robot movements with dynamic hand gestures.

III. Interfaces for Flight Path Generation

The user study described and analyzed throughout the remainder of this paper utilizes an adaptation of our previously developed gesture-based natural language interface [5] and a mouse-based interface [11]. Each interface gives a user the ability to define a complex flight path by defining individual trajectory segments with a library of twelve gesture primitives (Fig. 2). After all the desired trajectory segments have been defined, both interfaces automatically define any additional parameters (e.g., transition velocities) to combine the segments into a complete flight path. In both systems, two assumptions are made in regards to the trajectory segment library: (1) the *Circle* segment is defined in a clockwise direction parallel to the ground and (2) the *Spiral* segment is defined in the upward, clockwise direction parallel to the ground.

Fig. 3: System setup for flight path generation. The user is currently using the gesture interface.

A. Gesture-based Interface

In our previous work we implemented a complete end-to-end ground control system which contained five modules: volume definition, gesture, trajectory generation, validation, and flight [5]. The gesture-based interface used in this work is a self-contained variation which includes only the gesture, trajectory generation and validation modules. In the gesture-based interface a simplistic setup requires only two components: (1) a computer for running the interface and displaying feedback to the user and (2) a Leap Motion (Leap) controller (SDK v2.2.6) to track the gesture input of the user. The Leap uses three infrared cameras to track hand gestures with

Fig. 4: The Yes/No message window shown after defining each trajectory segment with the gesture interface [11].

sub-millimeter accuracy at 200 frames per second within an 8ft^3 interactive volume above the controller [27][28]. It provides an alternate input modality for users to naturally and intuitively define each primitive by mimicking their shape. As part of the system setup the Leap is placed on a surface in front of the user (Fig. 3). The current instantiation of the gesture interface assumes the user is right handed.

1) Gesture Module: The first module in the gesture-based interface is the Gesture Module. As part of this module the user's hand gesture is characterized as one of the twelve classes seen in the trajectory segment library (Fig. 2) using a trained Support Vector Machine (SVM) classifier. Once the gesture is classified the system displays a picture of the chosen trajectory segment as visual feedback. After each segment is defined a message window asks if they would like to define another segment. Performing a *Right* gesture indicates they would like to add another segment, whereas a *Left* gesture means they are finished and would like to see the complete flight path have have just built (Fig. 4).

The SVM classifier was trained with a linear kernel using data collected from eleven users. Each user provided ten samples per trajectory segment in the library (total of 120 data samples per user). The hand direction movement during the gesture and the eigenvalues of the hand position are used as features for classification.

2) Trajectory Generation Module: After the user has defined all desired trajectory segments the system automatically combines them into a flyable path for the UAV. This is accomplished by first creating a set of fifth order Bézier curves for each trajectory segment. Eq. 1 shows the general equation for a Bézier curve. They are polynomial over a finite interval, t, and expressed as a sum of Bernstein polynomials multiplied by a control point, where p represents a control point, n represents the degree of the polynomial and $0 \leq t \leq 1$ [29]. Each set is then connected in series, ensuring smoothed transition points.

$$p(t) = \binom{n}{i} \sum_{i=0}^{n} (1-t)^{n-i} t^i p_i \qquad (1)$$

3) Validation Module: In this gesture interface once the combined flight path is created using the Trajectory Generation Module, the Validation Module displays a visual representation to the user. This pictorial representation of the complete flight path gives a 3D view from the viewpoint of the user. As this interface implementation is meant for

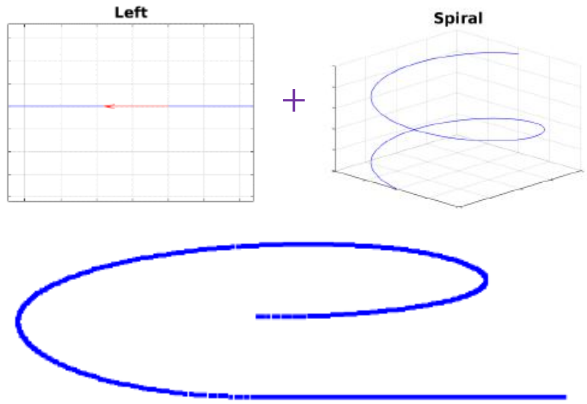


Fig. 5: Sample combined flight path generated by the Validation Module. Individual trajectory segments (above) are combined into a flyable path (below).

evaluation purposes only and no data is sent to a vehicle controller, no confirmation is needed from the user. This module is used as a simple method for feedback to the user on the accuracy of their trajectory segment definition. Figure 5 gives an example combined flight path shown by the Validation Module. In this example, the user defines a *Left* segment followed by a *Spiral*.

B. Mouse-based Interface

The mouse-based interface consists of a drop-down menu (Fig. 6) for choosing a desired trajectory segment from the given library (Fig. 2) [11]. A simple message window with buttons is used for the Yes/No message window between building trajectory segments (Fig. 7). All user feedback seen in the gesture interface is mimicked in the mouse-based interface. This interface assumes that a user will not define the same shape twice.

IV. Experimental Setup

As part of the user study conducted 13 subjects were asked to use both the gesture-based interface and a mouse-based interface. All subjects were allowed to sit or stand while using an interface. Due to the current instantiation of the gesture-based interface's assumption that the user would be performing gestures with their right hand, all subjects were asked their comfort level with using their right hand prior to the experiment. Only subjects who were right handed or left handed and comfortable with using their right hand were asked to participate. Subjects rated their right hand comfort level as part of the background questionnaire.

The order of interface use was counterbalanced across all subjects. In each set of trials a subject was asked to build a set of three flight paths using both the gesture and mouse interfaces (Fig. 8). The flight path order was randomized and counterbalanced, however each subject used the same order for both the gesture and mouse interface runs. Each flight path included three trajectory segments. Although the flight paths ranged in difficulty level to build, a *Right* segment was always included to avoid biases in segment ordering.

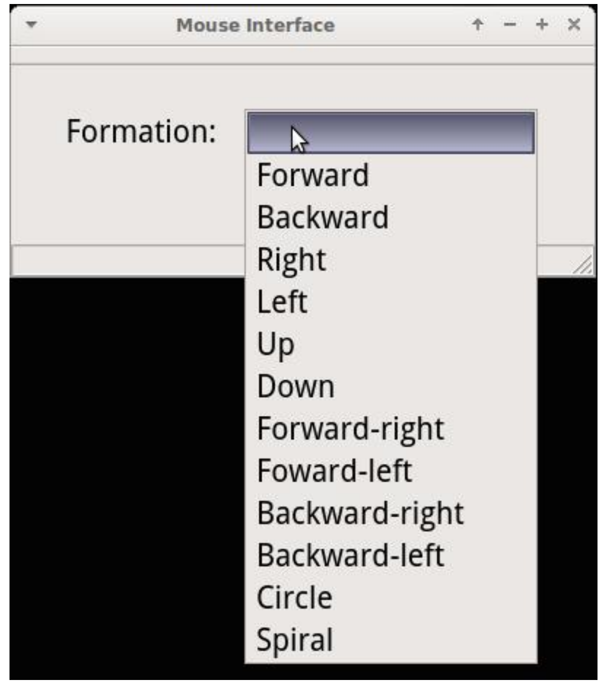


Fig. 6: Mouse interface drop-down menu for defining a desired trajectory segment.

Fig. 7: Message window shown after defining each trajectory segment with the mouse interface.

For each user study the researcher used the following protocol order: (1) subject reads and signs Privacy Act Notice and Informed Consent Form, (2) researcher(s) outline user study purpose and goals, (3) subject completes background questionnaire, (4) subjects train on interface, (5) subject builds given flight paths, (6) subject completes subjective questionnaire and NASA Task Load Index (TLX), and (7) steps 4-6 are repeated for second interface [30][31]. During training subjects were given a printout of the trajectory segment library (Fig. 2) and were allowed to keep the printout during the test runs. Before each test run, the subject was given a printout of desired flight path (one of the three shown in Fig. 8). Subjects were able to study the printout for only five seconds before the test run began. However, they were allowed to keep the printout throughout the entire duration of the run. The printouts contained the three labelled and numbered (in desired order) trajectory segments to be defined (Fig. 8). In addition to the data collected from the background questionnaire, NASA TLX, and subjective

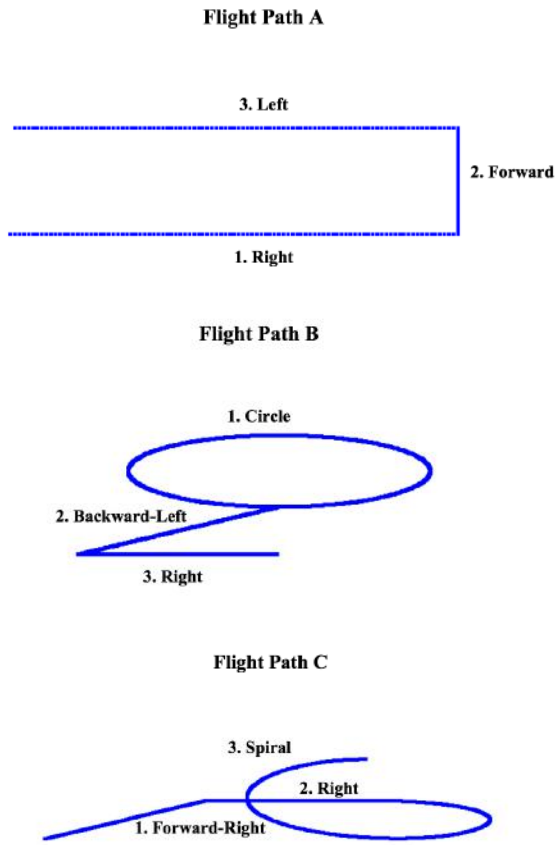


Fig. 8: The three flight paths subjects were asked to build. Subjects were given individual printouts for each including numbered and labelled segments [11].

questionnaires for each interface, researchers collected the following: (1) the time to complete each test run, (2) whether a subject chose to sit or stand while using an interface, and (3) the correctness of each flight path. All three additional sets of data were taken through observation.

A flight path is considered correct if all three desired trajectory segments are defined. The errors seen throughout the user study have been categorized as one of the following five errors: (1) system misinterprets correct human input, (2) an extra segment was added (in addition to the three required in each flight path), (3) human error – wrong or missing gesture input, (4) combination error – system misinterpretation plus a human error (error type 1 plus error type 3), and (5) combination error – system misinterpretation plus an extra segment was added (error type 1 plus error type 2). By tracking common combined errors we can see which errors can lead to secondary errors.

V. RESULTS

All results shown here are taken from the data collected in the background questionnaire, two NASA TLX workload measures (one after using each interface), and two subjective questionnaires (one after using each interface). An analysis

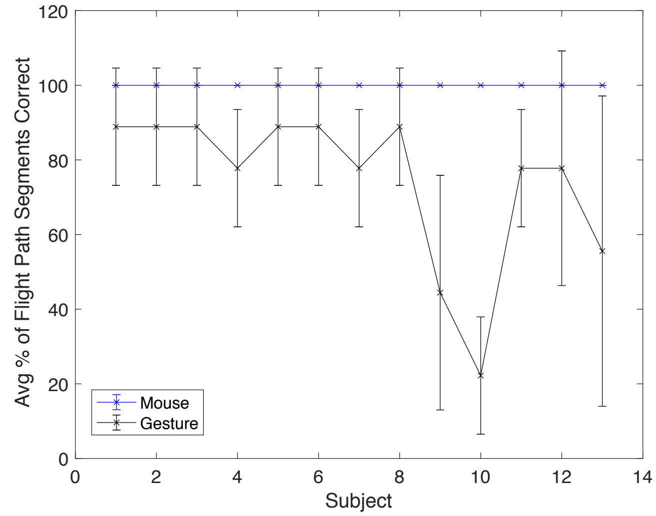


Fig. 9: The average percentage of each flight path that was defined correctly per subject with each interface.

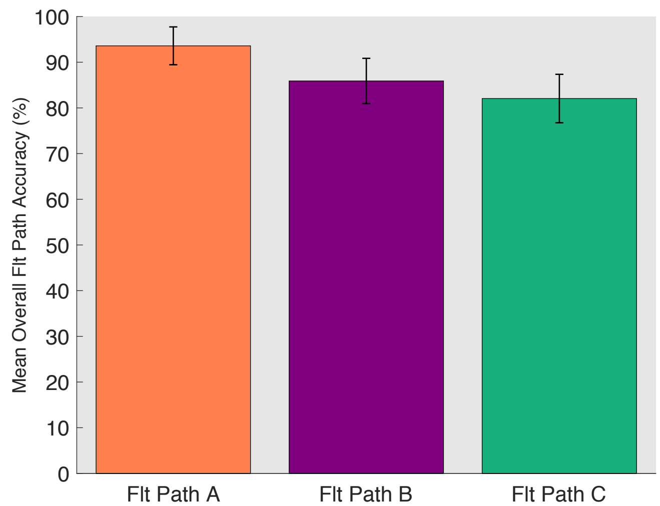


Fig. 10: The average percentage of each flight path that was defined correctly.

of variance (ANOVA) on the data was conducted using IBM SPSS version 24. The following independent variables were used to analyze between subject effects: (1) input interface (mouse or gesture), (2) previous experience flying UAVs, (3) right handed vs. left handed, (4) sit vs. stand and (5) flight path. In addition, the interaction between input and the other independent variables was analyzed. The results will show the effect of each variable on (1) the number of error segments, (2) overall flight path accuracy, (3) type of errors, (4) the time taken to build the given flight paths, and (5) subjective workload measures in the NASA TLX – mental, physical, temporal, performance, effort, and frustration. A Tukey HSD Post-Hoc test was run on flight path when it was significant ($p \leq 0.05$). Where appropriate, graphs are shown with error bars for the standard error of mean.

Each NASA TLX asked a subject to rate their perceived workload measures on a scale from 0 to 10. For mental

TABLE I: SEGMENT DEFINITION ERRORS BY TYPE

	Mouse	Gesture
Misinterpret	0%	41.03%
Extra Segment	2.56%	5.13%
Human Error	0%	2.56%
Human + Misinterpret	0%	5.13%
Extra + Misinterpret	0%	5.13%

demand, physical demand, temporal demand, effort, and frustration level 0 indicated a low workload while 10 indicated a high workload. The 0 for (subject perceived) performance represented good performance and 10 meant poor performance.

After using each interface subjects were asked to fill out a subjective questionnaire which asked them to rate the following on a likert scale from 1 to 5: (1) overall difficulty in using each interface, (2) interface responsiveness, (3) liklihood of using the interface again, (4) the amount of practice time given and (5) the amount of time given to study each flight path before a trial run. A 1 in difficulty indicated the interface was very easy to use, whereas a 5 indicated it was very difficult to use. The 5 in responsiveness meant that the interface was too fast, compared to a 1 which was too slow. The 1 in liklihood expressed that the subject was not likely to use the interface again, as compared to a 5 where they were very likely to use it again. For both the amount of time given to practice and to study the flight path, a 5 indicated there was too much time given and a 1 meant there wasn't enough time given. Once each subject had used both interfaces they were asked to rate their preference between the mouse and gesture interface. A 1 meant that they preferred the mouse interface while a 5 indicated their preference for the gesture interface.

From the background questionnaire we see that 76.92% of subjects were right hand dominant. Although some subjects were left handed, all said they were comfortable using the right hand. 23.08% had previous experience flying UAVs. For those who had previously flown UAVs, an average of 170.67 flight hours were logged over an average of 3.75 years. 7.69% of the subjects had previously used a gesture-based interface other than a cell phone or tablet [11].

A. Overall Segment Definition Accuracy

Overall, subjects defined 97.44% of flight paths correct while using the mouse interface and 41.03% correct when using the gesture interface. Excluding the error of adding an extra trajectory segment, the accuracy of defining the three desired individual segments with each interface was statistically significant with 100% of the trajectory segments defined by the mouse interface correct and 74.36% of the segments correctly defined with the gesture interface ($F_{(1,30)} = 79.510$, $p \leq 0.01$). Figure 9 displays the overall average percentage of flight paths that each subject correctly defined using the mouse versus the gesture interface. All but 2 subjects were able to correctly define more than 50% of flight paths with a majority of subjects defining more than

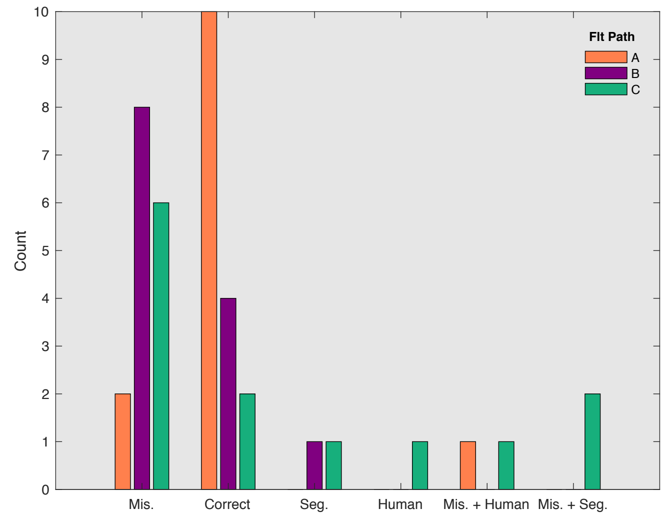


Fig. 11: Count of error per type when subjects built each flight path using the gesture interface.

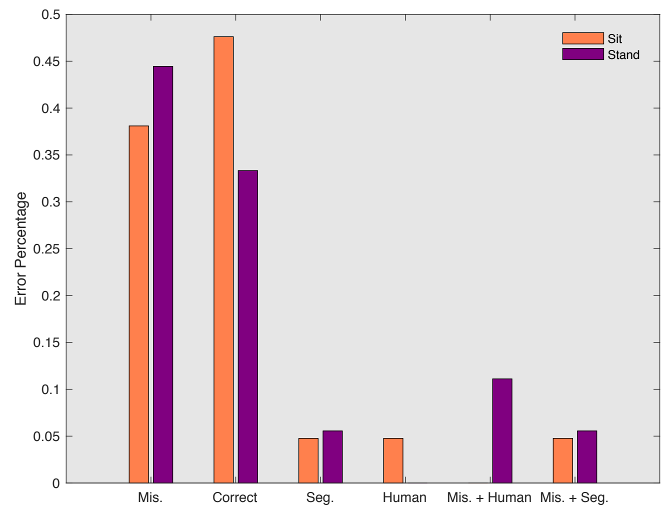


Fig. 12: Percentage of error type when subjects sat versus stood when using the gesture interface.

75% of flight paths correctly. Flight path C was the hardest to define at 82.05% correct, followed by flight path B and then A at 85.90% and 93.59% respectively (Fig. 10). This matched the difficulty of the gestures required as the *Spiral* was the hardest gesture to perform followed by the *Circle*.

B. Number of Error Segments

For each complete flight path defined, the number of error segments defined when a subject used the mouse-based interface ($M = 0$, $SE = 0$) was statistically less than when subjects used the gesture-based interface ($M = 0.77$, $SE = 0.14$) with $F_{(1,30)} = 79.510$, $p \leq 0.01$. The number of error segments seen in flight path A was significantly different than in flight path C. Right hand dominant subjects had a statistically significant lower number of error segments than those who were left hand dominant ($F_{(1,30)} = 10.294$, $p \leq 0.01$). The number of error segments seen from subjects

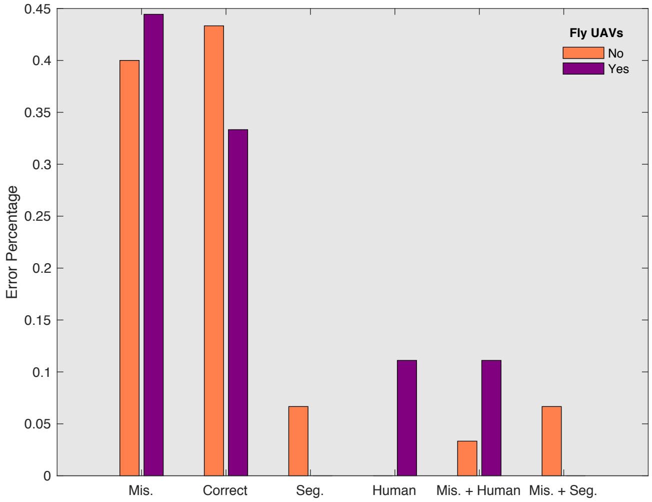


Fig. 13: Percentage of each error type and correct trajectory segments seen for subjects using the gesture interface given their previous UAV flight experience.

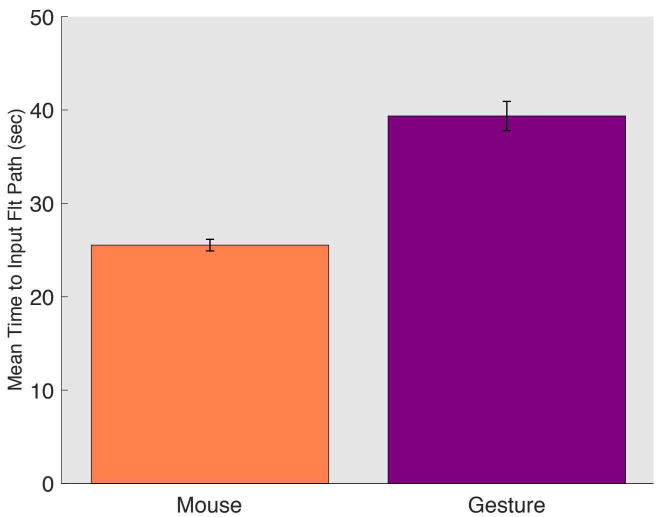


Fig. 14: The average time subjects took to build flight paths using each interface.

who stood during the trials was significantly higher than those who sat while using the interfaces ($F_{(1,30)} = 8.750$, $p \leq 0.01$).

C. Error Types

Table I displays the percentage of correct and error types seen when defining trajectory segments using each interface. A majority of errors seen from the gesture interface are attributed to the system misinterpreting an input gesture from the subject. The least number of errors seen when using the gesture interface came from human error – the subject performing the wrong gesture or defining fewer than the desired number of trajectory segments. All errors seen when using the mouse interface are a result of the subject adding an extra segment to the end of the desired flight path.

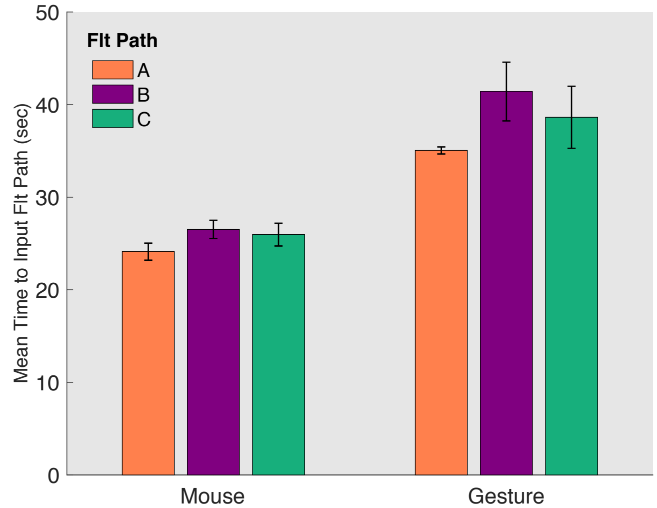


Fig. 15: The average time subjects took to build all three flight paths with each input interface.

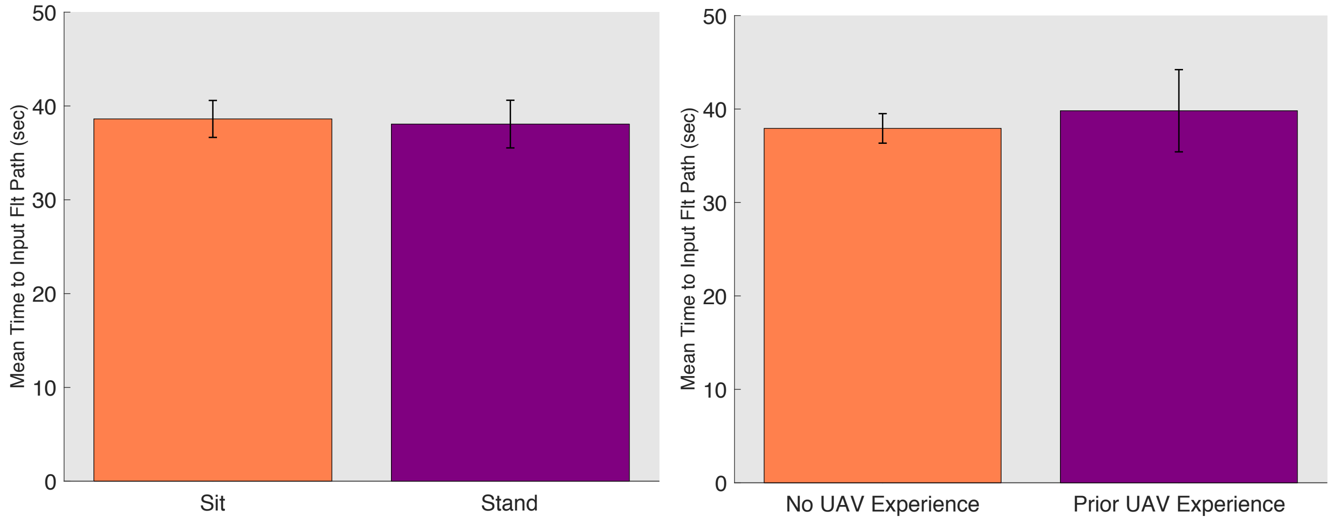
Of the errors seen when subjects used the gesture interface, a majority resulted from flight path C (Fig. 11). When building flight path C subjects were more likely to have the system misinterpret their hand gesture input. The least number of total errors were seen when subjects were building flight path A. Figure 12 shows that subjects who sat while using the gesture interface had more correct trajectory segments than errors when building segments. No errors were seen from purely human error when subjects stood. Subjects who had previous experience flying UAVs had no errors from adding unwanted additional segments to the flight path, but had a higher number of misinterpretations by the system (Fig. 13).

D. Time to Build Flight Path

The average time to build a flight path when using the mouse-based interface versus the gesture-based interface was statistically significant ($F_{(1,30)} = 80.474$, $p \leq 0.01$). Although the average time was less when using the mouse interface, the difference was less than 13 seconds (Fig. 14). Figure 15 shows that flight path B took longer to build than Flight paths A and C for both interfaces. The trend overall was statistically significant ($F_{(2,30)} = 5.001$, $p = 0.013$). Flight path A took the least amount of time to build on average for both interfaces. The time to build flight path A was statistically different than the time to build flight path B at the $p = 0.05$ level. Figure 16a shows there was little difference seen in the time to build flight paths for subjects who chose to sit versus stand ($M = 32.37$ seconds, $SE = 1.65$ and $M = 31.67$ seconds, $SE = 1.51$ respectively). The difference in time required to build flight paths using the gesture interface given their prior experience flying UAVs (Fig. 16b) was statistically significant ($F_{(2,15)} = 5.118$, $p = 0.039$).

E. Subjective Measures

Table II gives the average NASA TLX workload measure ratings given by subjects after using both the mouse and



(a) (b)

Fig. 16: The time (seconds) that subjects took on average to build the desired flight paths using the gesture interface.

TABLE II: AVG. NASA TLX MEASURES (from 0-10)

	Mouse	Gesture
Mental	1.92	4.77
Physical	0.85	3.50
Temporal	2.27	2.92
Performance	1.54	5.62
Effort	1.42	4.23
Frustration	1.46	4.62

gesture-based interface. The ratings for mental demand, physical demand, performance, effort, and frustration are statistically significant ($F_{(1,2)} = 15.583$, $p \leq 0.01$; $F_{(1,2)} = 10.924$, $p \leq 0.01$; $F_{(1,2)} = 134.000$, $p \leq 0.01$; $F_{(1,2)} = 15.044$, $p \leq 0.01$; and $F_{(1,2)} = 7.644$, $p = 0.02$ respectively). Subjects who were right hand dominant indicated a significantly lower effort ($F_{(1,2)} = 32.00$, $p = 0.03$).

Table III displays the NASA TLX workload measures for the gesture interface in more detail given prior experience flying UAVs, whether a subject chose to sit or stand, and their hand dominance. Subjects who had previous experience flying UAVs and chose to stand felt they performed better and had a lower workload in all measures except for Frustration than subjects who did not have previous UAV flight experience or chose to sit. Right hand dominant subjects' workload was lower for all measures. They also perceived their performance to be better.

Overall subjects thought the mouse interface was pretty easy to use ($M = 1.15$) as compared to an almost neutral difficulty level of the gesture interface ($M = 3.31$). In general subjects thought both interfaces were on the slow side ($M_{mouse} = 2.08$ and $M_{gesture} = 2.69$). Although subjects said they were more likely to use the mouse interface again in the future than the gesture ($M_{mouse} = 4.23$ and $M_{gesture} = 2.85$ respectively), their overall preference for the mouse interface was much closer to neutral ($M = 3.77$). For both interfaces, subjects felt that the right amount of time was given for training and studying the flight paths.

VI. Discussion

Analysis shows that although subjects were able to define a larger percentage of flight paths correctly using the mouse-based interface, a fairly high percentage of flight paths were still defined correctly using the gesture-based interface. As most subjects had no prior experience with gesture interfaces before the user study, this indicated that even with a limited amount of training and guidance the implemented gesture-based interface was relatively easy and intuitive to learn. In general subjects said they were more likely to use the mouse interface in the future than the gesture interface. However, given an almost neutral preference between the interfaces – albeit leaning towards the mouse interface – there appears to be an underlying acceptance of the gesture interface not reflected in the overall ratings.

The higher number of errors seen when building flight path C and the longer time used to define trajectory segments in flight path B emphasized the higher difficulty in performing the *Circle* and *Spiral* gestures. The *Spiral* gesture was more difficult than the *Circle* gesture as reflected in the reported accuracy of each flight path built. The difficulty of each flight path compared to the others (Fig. 11) did not correspond to the time required to build each flight path (Fig. 15).

When using the gesture interface, subjects with prior experience flying UAVs seemed more deliberate when defining segments. This resulted in a lower number of human errors (Fig. 13) and higher average time required to build flight paths (Fig. 16b). However, subjects without prior UAV experience had a higher proportion of correct segments to misinterpreted segments indicating their ability to learn the nuances of the gesture system faster than their experienced counterparts. Experienced UAV subjects' familiarity with

TABLE III: NASA TLX MEASURES FOR THE GESTURE-BASED INTERFACE

	No UAV Exp.	UAV Exp.	Sit	Stand	Left Handed	Right Handed
Mental	4.11	2.46	3.50	2.83	4.50	3.00
Physical	2.23	2.00	2.71	1.54	2.92	1.95
Temporal	2.73	2.12	3.25	1.83	3.67	2.23
Performance	3.63	3.42	3.86	3.25	3.58	3.56
Effort	3.05	2.08	2.93	2.71	4.00	2.48
Frustration	2.95	3.33	2.89	3.21	3.33	2.95

other interfaces intended for the same purpose as the gesture interface may account for this difference.

Some differences between subjects who chose to sit versus stand were seen in the number of error segments defined when comparing the interfaces. However, upon closer look of the gesture interface, neither condition lent itself to a significant difference in error types seen (Fig. 12) and an almost equal time was used to build flight paths (Fig. 16a).

Subjects seemed to have a more realistic impression of their skill when using the mouse interface as compared to the gesture interface. These differences were highlighted in the NASA TLX results. Overall, subjects rated their workload higher when using the gesture interface. They rated their perceived performance low to neutral more often when using the gesture interface even when they had defined a higher number of trajectory segments correctly. This suggests either (1) subjects are already familiar with their error rate when using the mouse interface compared to the gesture interface or (2) that subjects may be conflating the required additional training and difficulty during training on the gesture interface with their ability to use it after training.

On closer inspection of the workload measures for the gesture interface we see that although the average workload rating was higher than the mouse interface, these differences may be attributed to certain factors. Specifically, we find that prior experience flying UAVs, the choice to sit or stand, and hand dominance had a clear effect on workload ratings. As seen in Table III those with previous experience flying UAVs gave lower workload ratings and a better perceived performance than those without experience. Their familiarity with the intended use case most likely accounts for this difference. Additional training sessions to increase understanding of the mission requirements in the given use case may reduce the differences in the future.

Even though sitting versus standing had little significance the overall difference in performance between the interfaces, it did have a noticeable difference on the workload subjects felt when using the gesture interface (Tab. III). Subjects who chose to sit when using the gesture interface tended to feel a high overall workload than those who stood. The difference suggests that people might find their hand less constrained when standing as opposed to sitting, leading to more comfort and accuracy when performing gestures. Since all subjects were required to use their right hand when using the gesture interface, right hand dominant subjects unsurprisingly reported a lower workload than left handed subjects. Right hand dominant subjects also produced a fewer number of error segments. To increase the viability of the gesture interface, future system independence from the input hand side will be needed to decrease the highlighted workload differences.

Comments given in the questionnaires suggest that even with their better performance using the mouse interface, subjects were open to using the alternate gesture input modality. Several subjects noted that the current gesture interface may lead to user fatigue over time. This issue can be mitigated by implementing a method for users to define an entire flight path at once instead of piece-by-piece. It would also reduce the need for users to read the list of possible trajectory segments each time they wanted to define a new one. Subjects noted that the gestures themselves were intuitive and could be leveraged in the future to build complex flight paths that may otherwise be tedious to implement using a traditional mouse interface. However, they would like a method for modifying or correcting error segments. Although more practice would lead to an improved overall accuracy, subjects said that the innate assumptions made by the interface on the time each person would take to complete a gesture and when they would perform the gesture was the most challenging aspect. In addition, more feedback should be given to the user about when the system was expecting a gesture input versus when it was analyzing the previous input data.

VII. Conclusion and Future Work

This paper presented a novel gesture-based interface for defining a UAV mission using an unmounted sensor to build complex UAV movement with simplistic and intuitive dynamic hand gestures. A user study was conducted to examine the viability of the interface in the context of UAV flight path generation. A mouse-based interface was used for a baseline comparison. The results show that although the 13 subjects were able to define flight paths at a higher overall accuracy with the mouse interface than with the gesture interface, their preference between the interfaces was almost neutral. Subjects had little issue with the gestures themselves as compared to the limitations of the current gesture system instantiation. The intuitiveness of the gestures was highlighted by the fact that subjects were able to correctly define a high percentage of flight path segments correctly (74.36%) despite having a limited amount of training time. The *Circle* and *Spiral* gestures were the most challenging to perform. Prior experience flying UAVs did have an effect on accuracy, whereas the sitting versus standing had very little effect. Even when performing fairly well, subjects tended to rate their perceived performance worse than when using the mouse interface. In general the gesture-interface was accepted as an alternate input modality for UAV flight path generation and subjects believed the gestures themselves to be intuitive. However, future iterations should include more user feedback and more flexibility in performing the gestures themselves.

Given the results of the user study and feedback from subjects, future work should focus on extending the capability of the current gesture interface to provide users more flexibility. More broadly, the interface should also allow users to define additional parameters related to the UAV flight path such as distances, radii, and heights. This next generation system should leverage the strengths of the current gesture-based interface. As gestures can very quickly get tedious when used to define more specific parameters, an additional input modality will need to bridge the gap between general trajectory shapes and the desired fully defined vehicle flight path. One possible modality is speech. Since humans typically use a combination of speech and gesture to communicate with each other, a multimodal system which incorporates the strengths of both modalities might be a promising solution. Future work will explore the requirements of such a multimodal system and the challenges that arise.

Acknowledgment

The authors would like to thank Kyle McQuarry of Analytical Mechanics Associates, Inc. for his help in designing and creating the mouse interface.

References

[1] M. L. Wald, "Domestic drones stir imaginations, and concerns," *New York Times*, 2013.

[2] H. Chen, X.-m. Wang, and Y. Li, "A survey of autonomous control for uav," in *Artificial Intelligence and Computational Intelligence, 2009. AICI'09. International Conference on*, vol. 2. IEEE, 2009, pp. 267–271.

[3] G. Saggiani and B. Teodorani, "Rotary wing uav potential applications: an analytical study through a matrix method," *Aircraft Engineering and Aerospace Technology*, vol. 76, no. 1, pp. 6–14, 2004.

[4] D. Weatherington and U. Deputy, "Unmanned aircraft systems roadmap, 2005-2030," *Deputy, UAV Planning Task Force, OUSD (AT&L)*, 2005.

[5] M. Chandarana, A. Trujillo, K. Shimada, and B. D. Allen, "A natural interaction interface for UAVs using intuitive gesture recognition," in *Advances in Human Factors in Robots and Unmanned Systems*. Springer, 2017, pp. 387–398.

[6] B. D. Allen and N. Alexandrov, "Serious gaming for test & evaluation of clean-slate (ab initio) national airspace system (nas) designs," 2016.

[7] S. Wegener, S. Schoenung, J. Totah, D. Sullivan, J. Frank, F. Enomoto, C. Frost, and C. Theodore, "Uav autonomous operations for airborne science missions," in *AIAA 3rd" Unmanned Unlimited" Technical Conference, Workshop and Exhibit*, 2004, p. 6416.

[8] J. Reitsema, W. Chun, T. Fong, and R. Stiles, "Team-centered virtual interactive presence for adjustable autonomy," in *Space 2005*, 2005, p. 6606.

[9] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications," *Communications of the ACM*, vol. 54, no. 2, pp. 60–71, 2011.

[10] D. Perzanowski, A. C. Schultz, W. Adams, E. Marsh, and M. Bugajska, "Building a multimodal human-robot interface," *IEEE intelligent systems*, vol. 16, no. 1, pp. 16–21, 2001.

[11] M. Chandarana, E. Meszaros, A. Trujillo, and B. Allen, "Fly like this: Natural language interfaces for uav mission planning," in *Proceedings of the 10th International Conference on Advances in Computer-Human Interaction*. ThinkMind, 2017 (in press).

[12] "Put-that-there: Voice and gesture at the graphics interface," vol. 14, no. 3. New York, NY, USA: ACM SIGGRAPH Computer Graphics, 1980, pp. 262–270.

[13] M. Quigley, M. A. Goodrich, and R. W. Beard, "Semi-autonomous human-uav interfaces for fixed-wing mini-uavs," in *Intelligent Robots and Systems (IROS). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3. IEEE, 2004, pp. 2457–2462.

[14] N. Li, S. Cartwright, A. Shekhar Nittala, E. Sharlin, and M. Costa Sousa, "Flying frustum: A spatial interface for enhancing human-uav awareness," in *Proceedings of the 3rd International Conference on Human-Agent Interaction*. ACM, 2015, pp. 27–31.

[15] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 7, pp. 677–695, 1997.

[16] T. Naseer, J. Sturm, and D. Cremers, "Followme: Person following and gesture recognition with a quadrocopter," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 624–630.

[17] T. Ende *et al.*, "A human-centered approach to robot gesture based communication within collaborative working processes," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 3367–3374.

[18] W. S. Ng and E. Sharlin, "Collocated interaction with flying robots," in *2011 RO-MAN*. IEEE, 2011, pp. 143–149.

[19] J. R. Cauchard, J. L. E, K. Y. Zhai, and J. A. Landay, "Drone & me: an exploration into natural human-drone interaction," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2015, pp. 361–365.

[20] R. A. S. Fernández, J. L. Sanchez-Lopez, C. Sampedro, H. Bavle, M. Molina, and P. Campoy, "Natural user interfaces for human-drone multi-modal interaction," in *Unmanned Aircraft Systems (ICUAS), 2016 International Conference on*. IEEE, 2016, pp. 1013–1022.

[21] S. Iba, J. M. V. Weghe, C. J. Paredis, and P. K. Khosla, "An architecture for gesture-based control of mobile robots," in *Intelligent Robots and Systems, 1999. IROS'99. Proceedings. 1999 IEEE/RSJ International Conference on*, vol. 2. IEEE, 1999, pp. 851–857.

[22] P. Neto, J. N. Pires, and A. Moreira, "High-level programming for industrial robotics: using gestures, speech and force control," in *Submitted to the IEEE International Conference on Robotics and Automation (ICRA2009), Kobe, Japan*. Citeseer, 2009.

[23] S. Waldherr, R. Romero, and S. Thrun, "A gesture based interface for human-robot interaction," *Autonomous Robots*, vol. 9, no. 2, pp. 151–173, 2000.

[24] M. Becker, E. Kefalea, E. Maël, C. Von Der Malsburg, M. Pagel, J. Triesch, J. C. Vorbrüggen, R. P. Würtz, and S. Zadel, "Gripsee: A gesture-controlled robot for object perception and manipulation," *Autonomous Robots*, vol. 6, no. 2, pp. 203–221, 1999.

[25] J. Lambrecht, M. Kleinsorge, and J. Krüger, "Markerless gesture-based motion control and programming of industrial robots," in *Emerging Technologies & Factory Automation (ETFA), 2011 IEEE 16th Conference on*. IEEE, 2011, pp. 1–4.

[26] J. L. Raheja, R. Shyam, U. Kumar, and P. B. Prasad, "Real-time robotic hand control using hand gestures," in *Machine Learning and Computing (ICMLC), 2010 Second International Conference on*. IEEE, 2010, pp. 12–16.

[27] D. Bassily, C. Georgoulas, J. Guettler, T. Linner, and T. Bock, "Intuitive and adaptive robotic arm manipulation using the leap motion controller," in *ISR/Robotik 2014; 41st International Symposium on Robotics; Proceedings of*. VDE, 2014, pp. 1–7.

[28] Leap Motion, Inc., "Leap motion for mac and pc," 2017, retrieved: Jan. 2017. [Online]. Available: https://www.leapmotion.com/product/desktop

[29] R. Choe, J. Puig, V. Cichella, E. Xargay, and N. Hovakimyan, "Trajectory generation using spatial pythagorean hodograph bézier curves," in *AIAA Guidance, Navigation, and Control Conference*, 2015, p. 0597.

[30] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," *Advances in psychology*, vol. 52, pp. 139–183, 1988.

[31] J. Byers, A. Bittner, and S. Hill, "Traditional and raw task load index (tlx) correlations: Are paired comparisons necessary?" *Advances in Industrial Engineering and Safety*, pp. 481–485, 1989.