On the Performance Evaluation of a Vision-based Human-Robot Interaction Framework

Junaed Sattar
University of British Columbia
Vancouver, BC, Canada V6T 1Z4
junaed@cs.ubc.ca

Gregory Dudek
McGill University
Montreal, QC, Canada H3A 0E9
dudek@cim.mcgill.ca

ABSTRACT

This paper describes the performance evaluation of a machine vision-based human-robot interaction framework, particularly those involving human-interface studies. We describe a visual programming language called RoboChat, and a complimentary dialog engine which evaluates the need for confirmation based on utility and risk. Together, RoboChat and the dialog mechanism enable a human operator to send a series of complex instructions to a robot, with the assurance of confirmations in case of high task-cost or command uncertainty, or both. We have performed extensive human-interface studies to evaluate the usability of this framework, both in controlled laboratory conditions and in a variety of outdoors environments. One specific goal for the RoboChat scheme was to aid a scuba diver to operate and program an underwater robot in a variety of deployment scenarios, and the real-world validations were thus performed on-board the Aqua amphibious robot [4], in both underwater and terrestrial environments. The paper describes the details of the visual human-robot interaction framework, with an emphasis on the RoboChat language and the confirmation system, and presents a summary of the set of performance evaluation experiments performed both on- and off-board the Aqua vehicle.

1. INTRODUCTION

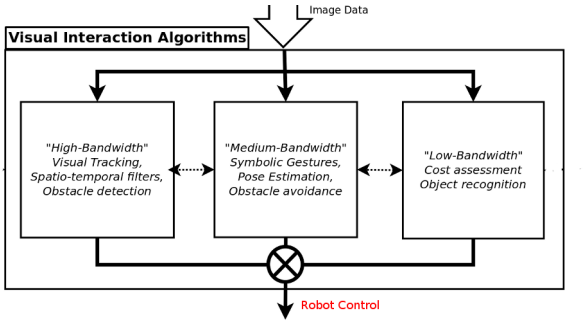
With the rapidly increasing adoption of robotic technologies in society, human-robot interaction frameworks and schemes are becoming more and more ubiquitous. "Traditional" interface devices and paradigms in computing and robotics are being replaced by more intuitive means of interaction, with "intuition" being broadly applied in the human interaction context. Speech, vision, tactile sensing etc. are but a few examples of such novel classes of interaction modalities. Our research explores the use of machine vision as a modality for human-machine interaction, building on the intuitive nature of visual gestures as a means of communication. In a wider scale, our vision-interaction framework also includes algorithms for person detection and tracking (in the underwater domain) and a learning-based tracker to robustly track objects with spatially complex color distributions. Algorithms of this nature, while directly not communicating with the human operator, assist mobile robots to exhibit "human-aware" behaviors, and are we label them as *implicit interaction* algorithms. The focus of this paper, however, will be on the more *explicit interaction* algorithms for human-robot interaction. Specifically, we look at a visual programming language for mobile robot programming called *RoboChat* [5] and a dialog management algorithm that evaluates the need for interaction based on risk and uncertainty [21]. We focus on the algorithms and the experimental validations performed to evaluate the usability of these techniques. The experiments were designed to assess usability through timing and accuracy measurements across a variety of task scenarios, and also included qualitative feedback from users as they operated the Aqua underwater robot [4] using these methods. However, we limit our discussion to the quantitative user studies for this paper, and briefly discuss the various issues and experimental outcomes of the robot field trials.

Validation of our research has focused on the usability of the individual algorithms, and the HRI framework as a whole, as a collection of disparate algorithms. Quantitatively measuring performance of the individual components are mostly straightforward, as the experimental setups can be arranged off-board in a laboratory setting, under controlled environments. Under such circumstances, measurements can be obtained minimizing error and experiments can be repeated arbitrary number of times (except for those involving human participants). In field trials, specially those involving robots operating in challenging environments, as is the case of our underwater vehicle, such measurements can be exceedingly difficult. Several aspects contribute to this hardship, including but not limited to the constraints of human endurance, finding participants with required skill levels (which often are beyond those required by the norms of operational certification, such as those held by scuba divers), requirements of special equipments – both experimental and for measurements, inability to reproduce experimental conditions and the resulting lack of repeatability of obtained results, and cognitive loading of participants, often resulting in incomplete and/or biased measurements. Keeping such issues in mind, we turn our attention to reporting measurements of coarser scale – number of successful trials, application of novel commands or behaviors, distances traveled, and confirmations requested are to name a few of these metrics. Finer levels of quantitative measurements are reserved for off-board experiments held in controlled settings.

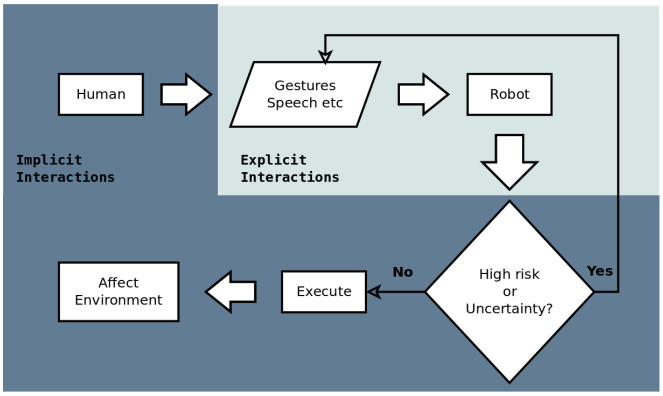
2. RELATED WORK

This paper presents, along with quantitative evaluations, of a human-robot interaction system that uses vision algorithms to communicate with and detect the presence of human operators (and other humans in the surrounding). Along with machine vision, this work spans the domains of gesture recognition, robot control, dialog management, and robot software architecture. In this Section,

PerMIS'12, March 20-22, 2012, College Park, MD, USA.



(a) Classification of algorithms in the described three-layer visual HRI framework.



(b) Explicit versus implicit interaction in the framework depicted in Fig. 1(a).

Figure 1: Core concepts of a vision-based HRI framework.

we present, albeit briefly, a summary of related previous work in these domains.

Our previous work looked at using visual communications, and specifically visual servo-control with respect to a human operator, to handle the navigation of an underwater robot [22]. In that work, the robot is able to follow a scuba diver, or any arbitrary target, to maneuver, but the diver accompanying the robot can only modulate the robot's activities by making hand signals that are interpreted by a second human operator sitting on a tethered robot control unit. Visual communication has also been used by several authors to allow communication between systems, for example in the work of Dunbabin *et al*. [6]

The work of Waldherr, Romero and Thrun [25] exemplifies the explicit communication paradigm in which hand gestures are used to interact with a robot and lead it through an environment. Tsotsos *et al*. [24] considered a gestural interface for non-expert users, in particular disabled children, based on a combination of stereo vision and keyboard-like input. As an example of implicit communication, Rybski and Voyles [17] developed a system whereby a robot could observe a human performing a task and learn about the environment. Such class of "learning-by-demonstration" tasks are part of a richly growing field, and are particularly attractive to human-robot interaction problems, where robot-human coexistence and coordination are of utmost importance.

Fiducial marker systems, as mentioned in the previous section, are efficiently and robustly detectable under difficult conditions. Apart from the ARTag toolkit mentioned previously, other fiducial marker systems have been developed for use in a variety of applications. The ARToolkit marker system [16] consists of symbols very similar to the ARTag flavor in that they contain different patterns enclosed within a square black border. The April Tag class of fiducials [13] also rely on square black-and-white markers, and have been used in vision-guided robotic tasks. Circular markers are also possible in fiducial schemes, as demonstrated by the Fourier Tags [18] fiducial system.

Gesture-based robot control has been considered extensively in Human-Robot Interaction (HRI). This includes explicit as well as implicit communication frameworks between human operators and robotics systems. Several authors have considered specialized gestural behaviors [9] or strokes on a touch screen to control basic robot navigation. Skubic *et al.* have examined the combination of several types of human interface components, with special emphasis on speech, to express spatial relationships and spatial navigation tasks [23].

Vision-based gesture recognition has long been considered for a variety of tasks, and has proven to be a challenging problem examined for over 20 years with diverse well-established applications [7][15]. The types of gestural vocabularies range from extremely simple actions, like simple fist versus open hand, to very complex languages, such as the American Sign Language (ASL). ASL allows for the expression of substantial affect and individual variation, making it exceedingly difficult to deal with in its complete form. For example, Tsotsos *et al.* [1] considered the interpretation of elementary ASL primitives (*i.e.*, simple component motions) and achieved 86 to 97 *per cent* recognition rates under controlled conditions. While such rates are good, they are disturbingly low for open-loop robot-control purposes.

While our current work looks at interaction under uncertainty in any input modality, researchers have investigated uncertainty modeling in human-robot communication with specific input methods. For example, Pateras *et al*. applied fuzzy logic to reduce uncertainty to reduce high-level task descriptions into robot sensor-specific commands in a spoken-dialog HRI model [14]. Montemerlo *et al*. have investigated risk functions for safer navigation and environmental sampling for the Nursebot robotic nurse in the care of the elderly [12]. Bayesian risk estimates and active learning in POMDP formulations in a limited-interaction dialog model [2] and spoken language interaction models [3] have also been investigated in the past. Researchers have also applied planning cost models for efficient human-robot interaction tasks [10] [11].

3. A FRAMEWORK FOR VISUAL HRI

In this work, an algorithm that enables a mobile robot to interact with a human, both through explicit and implicit communications, is labeled as an *Interaction Algorithm*. Algorithms belonging to the class of explicit interactions require an operator to give instructions to a mobile robot directly, for example through gestures or some direct input method. By using implicit interaction algorithms, a mobile robot can execute commands given by explicit instructions, particularly those that enable it to accompany the operator and assess task safety (from both the human and robot's perspective). Both classes of algorithms can further be categorized in a three-layer architecture, according to how frequently the functions are invoked by the robot. This three layer breakdown is demonstrated in Fig. 1(a), while a categorization of explicit versus implicit algorithms can be seen in Fig. 1(b). As we go from left-to-right in

Fig. 1(a), the rates of invocation for the algorithms in each box decreases; consequentially, the computation costs increase along the same directions, highlighting a natural inverse relationship between convocation rate and computational complexity.

The current implementation includes four major algorithmic components that facilitate vision-based human-robot interaction. These components are enumerated below:

1. A human-robot dialog model that evaluates the need for interaction based on utility and risk [21].
2. A visual language for programming robotic systems using gestures, and the human-interface studies towards quantifying its performance [5].
3. A visual biometric system for detecting and tracking multiple scuba divers in the underwater domain [20].
4. A machine learning algorithm to learn spatial color distribution of objects to achieve robust tracking under variable lighting and color distortion [19].

A comprehensive treatment of the framework is beyond the scope of this paper; instead, we limit the discussion on the core principles of the first two components, and present experimental setups and validation results.

3.1 Visual Programming

To visually program a robot, we use a set of engineered markers, called fiducials, to form simple geometric gestures that are interpreted by the robot as input commands. The underlying language, called *RoboChat*, enables the user to program the robot to carry out a large variety of tasks, both simple and complex in nature. RoboChat has a core set of basic tokens, including numerical digits, arithmetic operators, and relational operators. Additionally, RoboChat defines a limited number of variables, including command parameters, as well as some general-purpose variable names. RoboChat features two control flow constructs – the if-else statement, and the indexed iterator statement. The former construct allows the user to implement decision logic, while the latter immensely cuts down on the required number of tokens for repeated commands. The user can encapsulate a list of expressions into a numerically tagged macro, which can then be called upon later. This feature allows the reuse of code, which is essential when trying to minimize the number of tokens needed to specify behavior. Every construct is designed to minimize the number of tokens needed to express that construct. Reverse Polish notation (RPN) is heavily exploited to achieve this minimization – operators and operands are presented using RPN, eliminating the need for both an assignment operator and an end-of-command marker, while still allowing one-pass "compilation". Additionally, the use of RPN notation in the more abstract control flow constructs eliminate the need of various delimiters common to most programming languages. RoboChat interprets the tokens in real time, but only executes the commands upon detection of the `EXECUTE` token. This feature allows for batch processing, and also enables the error recovery element, using the `RESET` token.

As evident from the previous paragraph, the language is well-formed, governed by a strict grammar. This ability to instruct the robot with the aid of visual markers provides a first-order method for human-robot communication. By using fiducials, we also obviate the need for an error-free, robust gesture recognition algorithm. To compensate for errors in programming, RoboChat provides built-in syntax checking, and further error checking and uncertainty reduction is provided by a risk assessment engine, as described below.

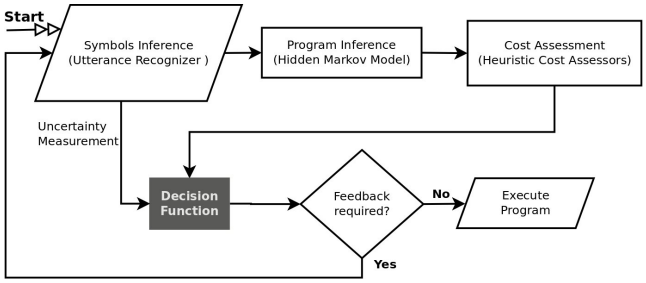


Figure 2: System flowchart for the confirmation dialog system.

3.2 Risk Assessment

In almost all real-world human-robot interfaces, there remains non-trivial uncertainty in input. If not accounted for in a robust manner, such uncertainty could lead to unsafe and potentially hazardous consequences for the robot and also cause harm to the operating environment. To minimize risk in the presence of uncertainty, we use a *Decision Function*, which takes into account belief states over a set of likely inputs, and the corresponding task execution costs. By using a Hidden Markov Model for belief tracking, and assessment of task costs through task simulation, the decision function requests confirmation of the high-risk input commands. That is, expensive tasks are executed if and only if they are truly requested by the user. An outline of the algorithmic flow for our system can be seen in Fig. 2.

4. USER STUDIES

We present a summary of user interface results for both RoboChat and the dialog system in this section. A substantial number of user interface studies were performed to evaluate the usability of these schemes, and an implementation of these systems are currently deployed on-board the Aqua family of underwater robots. The usability studies were performed across a wide range of users, and a number of representative tasks for our particular vehicle and its operating domains.

4.1 Experiments with RoboChat

We performed two sets of studies using the proposed marker-based input scheme in combination with the RoboChat language, to assess their usability. In both studies, the ARTag mechanism is compared to a hand gestures system, as competing input devices, particularly for environments unsuitable for the use of conventional input interfaces. The first study investigated the performance of the two systems with the user under significant stress, similar to the one scuba divers must face underwater. The second study compared the two input mechanisms in the presence of different vocabulary sizes. The main task in both studies is to input a sequence of action commands, with the possibility of specifying additional parameters, as accurately and efficiently as possible. The RoboChat format is used with both input devices, although in the case of the hand signal system, the gestures are interpreted by an expert human operator remotely, who subsequently validates the correctness of the input using the RoboChat syntax. This setup is realistic because in the case of our particular application, the diver's hand signals are interpreted by an operator on land, who then takes control of the robot. Also, the operator is not forced to be unbiased when interpreting gestures, because realistically the robot operator will guess and infer at what the diver is trying to communicate, if the hand gestures are ambiguously perceived.

1	TURN RIGHT, REVERSE, EXECUTE
2	FORWARD, TURN LEFT, FORWARD, EXECUTE
3	REVERSE, TURN RIGHT, FORWARD, REVERSE, EXECUTE
4	REVERSE, TURN RIGHT, REVERSE, TURN LEFT, REVERSE, EXECUTE
5	TURN RIGHT, SURFACE, TURN LEFT, SURFACE, REVERSE, TURN LEFT, STOP, EXECUTE
6	STOP, FORWARD, SURFACE, TURN RIGHT, SURFACE, EXECUTE
7	REVERSE, STOP, SURFACE, FORWARD, STOP, SURFACE, TURN RIGHT, EXECUTE
8	FORWARD, STOP, FORWARD, SURFACE, STOP, EXECUTE
9	TURN RIGHT, TURN LEFT, SURFACE, EXECUTE
10	FORWARD, REVERSE, FORWARD, STOP, EXECUTE
11	FORWARD, REVERSE, TURN RIGHT, EXECUTE

Table 1: Tasks used in Study A.

4.1.1 *Study A*

In the first study, the ARTag markers are provided to the participants, and they are allowed to place them in any configuration in the provided work area, particularly in a manner so that the tags can be easily accessible. The hand gestures in this study are predetermined, and are visually demonstrated to the participants, who are then asked to remember all the gestures. During the experiment session, the participants must rely on memory alone to recall the gestures, much like the case for the scuba divers.

The stress factor in the first study is introduced by asking participants to play a game of Pong (a classical 1970's table tennis video game [8]) during the experimental sessions. A suitable distractor task must be fairly accessible to all users, continually demanding of attention, yet still allow the core task to be achievable. Pong was decided to be closely fulfilling such requirements, and was chosen as a distractor task. This particular implementation of Pong uses the mouse to control the user's paddle. As such, participants are effectively limited to using only one hand to manipulate the markers and to make out gestures, while constantly controlling the mouse with the other hand. But since some of the predefined hand gestures require the use of both hands, this distraction introduces additional stress for the participants in terms of the alternatively showing gestures and playing Pong. Also, the experiment rules make it mandatory to inform the participant when the entered command is incorrect, and proceed onto the next command only after receiving the previous one correctly.

4.1.2 *Study B*

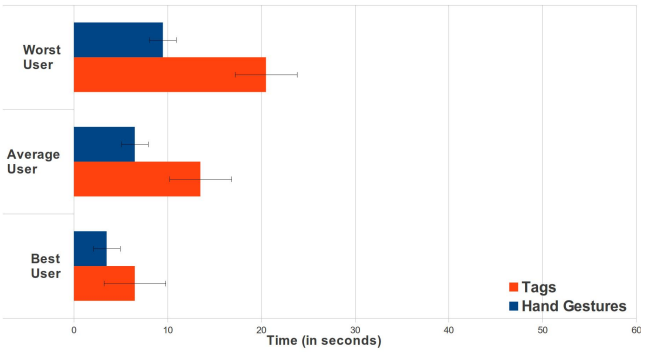
For the second study, the parameter of interest is the performance difference using different vocabulary sizes. Two vocabulary sets are used in this study – the first set contains only 4 action commands, while the second includes 32. This distinction is mentioned to every participant so that they can use this information to their advantage. As it is unrealistic to ask participants to remember more than 50 different hand gestures under the experiment's tight time constraints, a gesture lookup sheet is given to each participant. The subjects are encouraged to familiarize themselves with this cheat sheet during the practice sessions, to ensure that they spend minimal time searching for particular hand signals. The ARTag markers are also provided in the form of 'flip-books' to facilitate fast lookup and easy access. There is no distraction factor in this second study, but at the same time, the system accepts incorrect commands without informing the participants or making them re-enter the commands. The users are informed of this criterion, and are recommended to constantly keep track of the entered tokens and try to make as few mistakes as possible.

FORWARD, TURN LEFT, FORWARD, TURN RIGHT, REVERSE, STOP, FORWARD, SURFACE, TURN RIGHT, SURFACE, STOP, FORWARD, DEPTH, 10, TURN RIGHT, FORWARD, STOP, TAKE PICTURE, SURFACE, GPSFIX, DEPTH, 15, FORWARD, TURN RIGHT, FORWARD, TAKE PICTURE, 10, TURN LEFT, FORWARD, SURFACE, STOP, EXECUTE

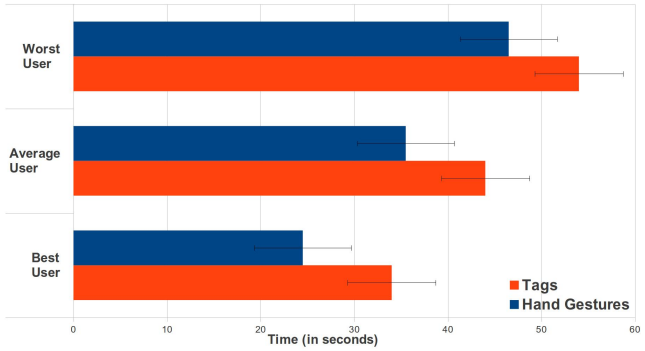
Table 2: Example of a long command used in Study B.

4.1.3 *Criteria*

Two criteria are used to compare the performance of the two input interfaces. The first criterion is speed; *i.e.*, the average speed it takes to enter a command. A distinction is made between the two studies regarding this metric: in the first study, the input time per command is measured from the time a command is shown on screen until the time the command is *correctly* entered by the participant, whereas in the second study, the command speed does not



(a) Study A: Average time taken per command using ARTag markers (in red) and using hand gestures (in dark blue).



(b) Study B: Average time taken per command using ARTag markers (in red) and using hand gestures (in dark blue).

Figure 3: Timing data for programs: Hand gestures vs RoboChat.

take into consideration the correctness of the command. The second study also uses the average time per individual token as a comparison metric. This metric demonstrates the raw access speeds of both input interfaces outside the context of RoboChat or any other specific environment.

The second criterion used to compare the two systems is the error rate associated to each input scheme. Once again, due to the distinction between how incorrect commands are treated between the two studies, results from this metric cannot be compared directly between studies. This criterion is used to look at whether the two schemes affect the user's performance in working with RoboChat differently.

In total, 12 subjects participated in study A, whereas 4 subjects participated in study B. One of the participants present in both studies has extensive experience with ARTag markers, RoboChat, and the hand gesture system. This expert user is introduced in the dataset to demonstrate the performance of a well-trained user. However, this user has no prior knowledge of the actual experiments, therefore is capable of exhibiting similar performance improvements throughout the sessions.

4.1.4 Results: Study A

One obvious observation we can make from the performance data is that the gesture system allows for faster communication than the marker system. The ratio between the two input techniques for some users surpasses 3:1 favoring hand gestures, while data from other users (including those from the expert user) show ratios of lower than 2:1. Since all users have experience with primitive hand gestures, we can infer that it may simply be that those users who did almost equally well with markers as gestures adapted to the marker system more quickly. Thus, the data suggest that the ARTag markers are capable of matching half the speed of the hand gestures, even given only limited practice. It is worth noting that contrary to the hand gestures which are chosen to have intuitive and natural mappings to their corresponding tokens, the mappings between the ARTag markers and tokens are completely arbitrary.

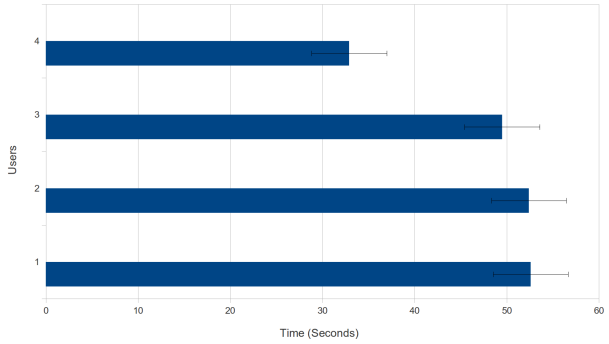
To further substantiate the hypothesis that the enhanced performance of hand gestures is due to familiarity, note that Fig. 3 indicates that the spread of the average time per command using gestures ($\pm$ 3 seconds) is much smaller than that for markers ($\pm$ 8 seconds). Arguably the more sporadic spread for the markers is due to unfamiliarity with this new input interface.

The distraction task (playing Pong) also plays an important role in increasing the performance disparity between the two systems. For each token, the participants need to search through the entire ARTag vocabulary set for the correct marker, whereas the associated hand gesture can be much easily recalled from memory. Since the Pong game requires the participant's attention on an ongoing basis, the symbol search process was repeatedly disrupted by the distraction task, amplifying the marker search time.

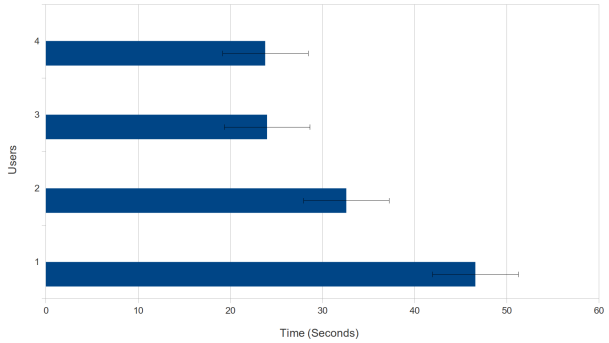
In terms of the error rate associated with each system, all the participants displayed error rates of roughly 5 *per cent* for both systems. This finding is surprising and interesting, because even though the symbolic system is harder to learn, it does not seem to generate more errors than the gesture system, even for inexperienced users.

4.1.5 Results: Study B

The data from study B suggests that the two input interfaces have very similar performances under the new constraints. Major contributing factors include the increase in the vocabulary size and the inclusion of many abstract action tokens (such as `RECORD_VIDEO` and `POWER_CYCLE`). This variation takes away the crucial advantage gestures had in the former study, and participants are now forced to search through the gesture sheet rather than remembering the many hand gestures. Essentially, in this study, the command speed criterion boils down to the search speed for each input device, and therefore depends on the reference structure, whether it is the ARTag flipbook or the gesture cheat sheet. And using the two engineered reference structures, the data of the experiments show that the speed performance of both input systems are actually very similar. Interestingly enough, the data spread between systems are actually reversed, as shown in Fig. 4. With the exception of the expert user, the average command and token speeds for all the participants using ARTag markers are almost identical, whereas the same speeds using gestures are now erratic between individuals. This result can be attributed to the fact that since the gestures are not kept in memory, different subjects adapt to the cheat sheet setup at different speeds.



(a) Study B: Average time taken per command across users using ARTag markers.



(b) Study B: Average time taken per command across users using hand gestures.

Figure 4: Study B: Average time taken per command using ARTag markers and hand gestures. In both plots, user 4 is the "expert user".

4.2 Experiments with the Dialog System

We performed a set of user studies to collect quantitative performance measures of our algorithm. When operating as a diver's assistant in underwater environments, the system uses fiducials to engage in a dialog with the robot. However, in the off-board bench trials, we employed a simplified "gesture-only language", where the users were limited to using mouse input. We used a vocabulary set of 18 tokens defined by oriented mouse gestures, and as such

each segment is bounded by a 20^o-wide arc. The choice for using mouse gestures stemmed from the need to introduce uncertainty in the input modality, while keeping the cognitive load roughly comparable to that experienced by scuba divers. We could not use the ARTag scheme for these experiments, as the ARTag library neither provides a confidence factor for tag detection, nor does it have a significant false positive detection rates. Using ARTags would have provided insufficient data to throughly validate the algorithm for an arbitrary input modality.

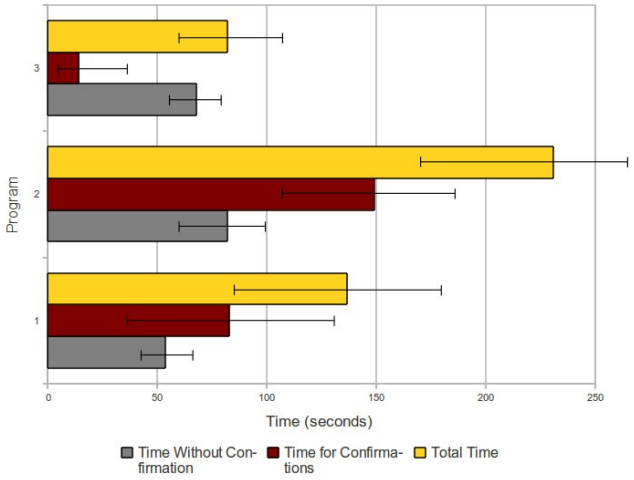
4.2.1 Experimental Setup

To calculate uncertainty in input, we trained a Hidden Markov Model using commonly used programs given to the robot (such as those used in previous experiments and field trials; *i.e.*, real operational data). To estimate task costs, we simulated the programs using a custom-built simulation engine and used a set of assessors that takes into account the operating context of an autonomous underwater vehicle. The simulator has been designed to take into account the robot's velocity, maneuverability and propulsion characteristics to accurately and realistically simulate trajectories taken by the robot while executing commands such as those used in our experiments. In choosing assessors for the user studies, we considered factors that directly affect underwater robot operations. For example, the distance traveled by the robot (and the farthest distance it travels from the start point) often has a direct bearing on the outcome of the mission, as the probability of robot recovery is inversely proportional to these factors. That is because energy consumption is directly proportional to the distance traveled. Robot safety (*e.g.*, chance of collisions) is also significantly compromised by traveling large distances. In particular, we applied four assessors during the user studies, which assessed total distance, furthest distance, execution time and average distance traveled.

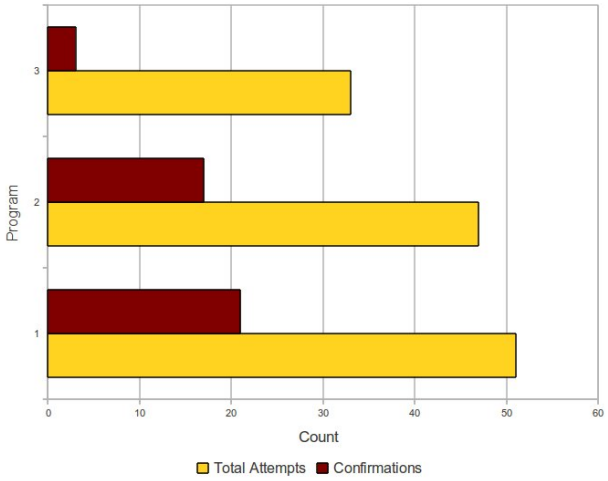
Each user was given three programs to send to the system, and each program was performed three times. A total of 10 users participated in the trials, resulting in 30 trials for each program, and 90 programs in all; Except for mistakes that created inconsistent programs, users did not receive any feedback about the correctness of their program. When a user finished writing a program, she either received feedback notifying her of program completion, or a confirmation dialog was generated based on the output of the Decision Function. The users were informed beforehand about the estimated cost of the program; *i.e.*, whether to expect to receive a feedback or not. In case of a confirmation request for Programs 1 and 3, the users were instructed to redo the program. For Program 2, the users were informed of the approximate values of the outputs of the assessors. In all cases, users were required to conduct the programming task until the output of the system (*i.e.*, either quantitative values from assessor outputs or confirmation dialogs) was consistent with the expected behavior. It is worth noting, however, that this does not necessarily indicate correctness of the programming, but merely indicates that the Decision Function has judged the input program (and likely alternatives of that) to be sufficiently safe (*i.e.*,"inexpensive") and thus safe for execution.

4.2.2 Results

From the user studies, it was observed that in cases where the programs were correctly entered, the system behaved consistently in terms of confirmation requests. Program 2 was the only one that issued confirmations, while Programs 1 and 3 only confirmed that the task would be executed as instructed. As mentioned, the users were not given any feedback in terms of program correctness. Thus, the programs sent to the robot were not accurate in some trials; *i.e.*, the input programs did not match exactly the programs given to the users. In case of mistakes, the Decision Function evaluated the input program and most likely alternatives, and only allowed a program to be executed (without confirmation) if and only if the task was evaluated to be less costly.



(a) Programming times, all users combined.



(b) Programming attempts and generated confirmations, all users combined.

Figure 5: Results from user studies, timing 5(a) and confirmations 5(b).

The cost of feedback, not unexpectedly, is the required time to program the robot. As seen in Figure 5(a), all three programs took more time to program on average with confirmations (top bar in each program group). From the user studies data, we see that the use of confirmations increases total programming time by approximately 50%. Although the users paid a penalty in terms of programming time, the absence of safety checks meant a greater risk to the system and higher probability of task failures. This was illustrated in all cases where the system issued a confirmation request; an example of which is demonstrated in a trial of program 3 by user 2. The input to the system was given as

"`LEFT` **`9`** `RIGHT 3 MOVIE 3` **`FOLLOW`** `FOLLOW` **`9`** `UP GPSFIX EXECUTE`"

where the mistakes are in bold. The system took note of the change in duration from $\mathbf{6 \times 3 = 18}$ seconds to $\mathbf{9 \times 3 = 27}$ sec-

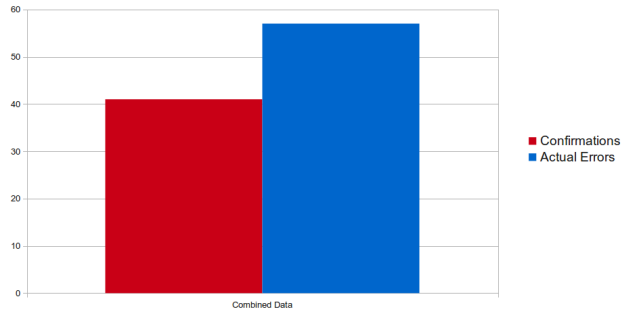


Figure 6: Error filter rate plot over all user studies data.

onds on two occasions, but more importantly, the `FOLLOW` command was issued without a `TUNETRACKER` command. This, and the change in parameters to the higher values, prompted the system to generate a confirmation request, which helped the user realize that mistakes were made in programming. A subsequent reprogramming fixed the mistakes and the task was successfully accepted without a confirmation. The distribution of confirmation requests and total number of attempts to program is shown in Figure 5(b).

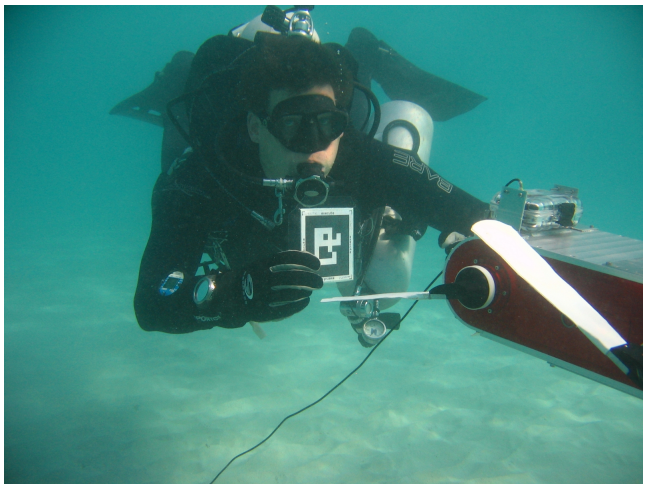
To further establish the benefits of this approach, we introduce a metric termed the *Error Filter Rate* (EFR). The EFR is a measure of the number of confirmations compared to the number of mistakes made by users during a programming task; *i.e.*,

$$\text{EFR} = \frac{\text{Confirmations}}{\text{Total Errors}}$$

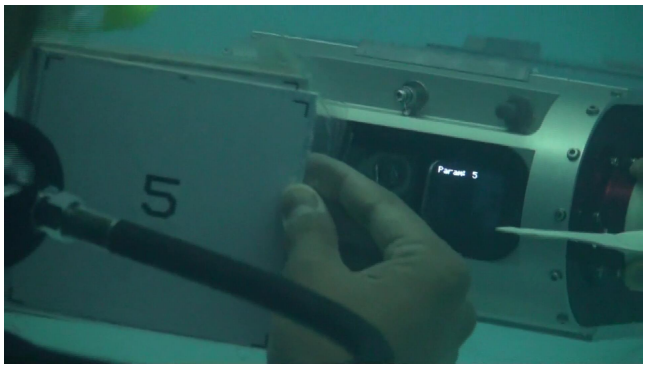
The EFR indicates the percentage of erroneous inputs which the system deemed to be dangerous; in other words, a low EFR value does not necessarily indicate a low error rate in programming, but indicates that most of the commands to the robot are interpreted as low-risk. In our studies, we achieved an EFR of approximately 72.8 *per cent*, as can be seen in Fig. 6, indicating the system interpreted roughly 72% of the erroneous commands as high-risk and intervened (with confirmation dialogs) to ensure the user's true desire.

5. FIELD TRIALS

We performed field trials of our system on-board the Aqua underwater robot, in both open-ocean and closed-water (controlled))environments. In both trials, the robot was visually programmed using RoboChat with the same language set used for the user studies, with ARTag and ARToolkitPlus [16] fiducials used as input tokens. The assessors used for the dialog user studies were also used in the field trials; in addition, we provided an assessor to take into account the depth of the robot during task execution. Because of the inherent difficulty in operating underwater (as discussed in Sec. 1), the trials were not timed. Users were asked to do each program once. Unlike in the user study, where there was no execution stage, the robot performed the tasks that it was programmed to do, when given positive confirmation to do so. In all experimental cases, the robot behaved consistently, asking confirmations when required, and executing tasks immediately when the tasks were inexpensive to perform. Unlike the user study, where the users had no feedback, the field trial participants were given limited feedback in the form of symbol acknowledgement using an micro-Organic-LED (Light Emitting Diode) or μOLED display at the back of the robot. Also unlike the user studies, the field trial participants were given access to a command to delete the program and start from the beginning, in case they made a mistake. A pictorial demonstration of our system in action during field trials can be seen in Fig. 7, which demonstrates the visual programming, and command feedback through the μOLED screen.



(a) A diver programming Aqua during ocean trials. The trailing cable is for a floating GPS antenna buoy on the ocean surface.



(b) Example of command acknowledgement given on the LED screen of the the Aqua robot during field trials.

Figure 7: Field trials of the proposed algorithm on board the Aqua robot.

6. CONCLUSIONS

This paper describes the experimental validations of two algorithms for *explicit* visual human-robot interaction which are components of a larger visual-HRI framework. The results focus primarily on the user studies, and also discusses key observations and findings from our field trials. From the results, it can be seen that individually both RoboChat and the dialog system increase efficiency and robustness in human-robot communication, particularly in areas where more traditional means of communication is not viable. The combination of both, however, is the most effective mean to increase fault-tolerance arising from mistakes in instructions, and communication uncertainty. RoboChat provides users with a expressive yet compact method for instructing a mobile robot, and the dialog engine, in a complimentary manner, ensures task and user safety, which is a much sought-after design goal of HRI systems. The implicit interaction algorithms, though not discussed for the sake of topic coherence, helps to create an effective scheme to

compliment these explicit interaction mechanisms.

Future research will aim to quantify system performance in real-world scenarios. As we discussed in the paper, a number of significant issues prevent accurate measurements of performance metrics in the field, particularly those that relate to human-centric systems. An open issue is the trade-off between expressivity, ease of use, flexibility and the minimization of coding errors. It seems that for different applications, different language subsets may be best and, in fact, this is what we have sometimes done in some practical deployments. One of our future goals is to design instrumentation capabilities as part of the framework itself, such that measurements of human operational data (along with robot performance) happens in an integrated manner. By design, the framework should be applicable to arbitrary robots across a variety of operating domains, irrespective of actual algorithms or modalities being used for human interaction. This remains a difficult challenge, and an open problem. Future goals also include adapting robot behaviors based on user input and feedback, such that more streamlined dialogs can be presented to the user, for example. As before, the challenge remains in appropriately quantifying operational parameters in a human context, such that operator preference and robot capabilities can be closely correlated towards creating a seamless man-machine interface.

7. REFERENCES

[1] K. Derpanis, R. Wildes, and J. Tsotsos. Hand gesture recognition within a linguistics-based framework. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 282–296, 2004.

[2] F. Doshi, J. Pineau, and N. Roy. Reinforcement learning with limited reinforcement: Using Bayes risk for active learning in POMDPs. In *Proceedings of the 25th international conference on Machine learning*, pages 256–263. ACM New York, NY, USA, 2008.

[3] F. Doshi and N. Roy. Spoken language interaction with model uncertainty: an adaptive human-robot interaction system. *Connection Science*, 20(4):299–318, 2008.

[4] G. Dudek, M. Jenkin, C. Prahacs, A. Hogue, J. Sattar, P. Giguère, A. German, H. Liu, S. Saunderson, A. Ripsman, S. Simhon, L. A. Torres-Mendez, E. Milios, P. Zhang, and I. Rekleitis. A visually guided swimming robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3604–3609, Edmonton, Alberta, Canada, August 2005.

[5] G. Dudek, J. Sattar, and A. Xu. A visual language for robot control and programming: A human-interface study. In *Proceedings of the International Conference on Robotics and Automation ICRA*, pages 2507–2513, Rome, Italy, April 2007.

[6] M. Dunbabin, I. Vasilescu, P. Corke, and D. Rus. Data muling over underwater wireless sensor networks using an autonomous underwater vehicle. In *International Conference on Robotics and Automation, ICRA 2006*, Orlando, Florida, May 2006.

[7] R. Erenshteyn, P. Laskov, R. Foulds, L. Messing, and G. Stern. Recognition approach to gesture language understanding. In *13th International Conference on Pattern Recognition*, volume 3, pages 431–435, August 1996.

[8] S. L. Kent. *The ultimate history of video games: from Pong to Pokémon and beyond : the story behind the craze that touched our lives and changed the world*. Prima, 2001.

[9] D. Kortenkamp, E. Huber, and R. P. Bonasso. Recognizing and interpreting gestures on a mobile robot. In *Proceedings of the thirteenth national conference on Artificial intelligence - Volume 2*, AAAI'96, pages 915–921. AAAI Press, 1996.

[10] K. Krebsbach, D. Olawsky, and M. Gini. An empirical study of sensing and defaulting in planning. In *Artificial intelligence planning systems: proceedings of the first international conference, June 15-17, 1992, College Park, Maryland*, page 136. Morgan Kaufmann Pub, 1992.

[11] D. Kulic and E. Croft. Safe planning for human-robot interaction. In *2004 IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04*, volume 2, 2004.

[12] M. Montemerlo, J. Pineau, N. Roy, S. Thrun, and V. Verma. Experiences with a mobile robotic guide for the elderly. In *Proceedings of the 18th National Conference on Artificial Intelligence AAAI*, pages 587–592, 2002.

[13] E. Olson. AprilTag: A robust and flexible visual fiducial system. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3400–3407, May 2011.

[14] C. Pateras, G. Dudek, and R. D. Mori. Understanding referring expressions in a person-machine spoken dialogue. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, 1995. (ICASSP '95)*, volume 1, pages 197–200, May 1995.

[15] V. 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*, 19(7):677–695, 1997.

[16] I. Poupyrev, H. Kato, and M. Billinghurst. *ARToolkit User Manual Version 2.33*. Human Interface Technology Lab, University of Washington, Seattle, Washington, 2000.

[17] P. E. Rybski and R. M. Voyles. Interactive task training of a mobile robot through human gesture recognition. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 664–669, 1999.

[18] J. Sattar, E. Bourque, P. Giguère, and G. Dudek. Fourier tags: Smoothly degradable fiducial markers for use in human-robot interaction. In *Proceedings of the Fourth Canadian Conference on Computer and Robot Vision*, pages 165–174, Montréal, QC, Canada, May 2007.

[19] J. Sattar and G. Dudek. Robust servo-control for underwater robots using banks of visual filters. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, pages 3583–3588, Kobe, Japan, May 2009.

[20] J. Sattar and G. Dudek. Underwater human-robot interaction via biological motion identification. In *Proceedings of the International Conference on Robotics: Science and Systems V, RSS*, pages 185–192, Seattle, Washington, USA, June 2009. MIT Press.

[21] J. Sattar and G. Dudek. Towards quantitative modeling of task confirmations in human–robot dialog. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, pages 1957–1963, Shanghai, China, May 2011.

[22] J. Sattar, P. Giguère, G. Dudek, and C. Prahacs. A visual servoing system for an aquatic swimming robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1483–1488, Edmonton, Alberta, Canada, August 2005.

[23] M. Skubic, D. Perzanowski, S. Blisard, A. Schultz, W. Adams, M. Bugajska, and D. Brock. Spatial language for human-robot dialogs. *IEEE Transactions on Systems, Man and Cybernetics, Part C*, 34(2):154–167, May 2004.

[24] J. K. Tsotsos, G. V. S. Dickinson, M. Jenkin, A. Jepson, E. Milios, F. Nuflo, S. Stevenson, M. B. adn D. Metaxas, S. Culhane, Y. Ye, , and R. Mannn. PLAYBOT: A visually-guided robot for physically disabled children. *Image Vision Computing*, 16(4):275–292, April 1998.

[25] S. Waldherr, S. Thrun, and R. Romero. A gesture-based interface for human-robot interaction. *Autonomous Robots*, 9(2):151–173, 2000.