A Comparison of Smartphone Interfaces for Teleoperation of Robot Arms

Diego Rodriguez
Department of CSE
Universidad de los Andes
Bogota, Colombia
dh.rodriguez25@uniandes.edu.co

Camilo Perez
Department of CS
University of Alberta
Edmonton, Canada
caperez@ualberta.ca

Martin Jagersand
Department of CS
University of Alberta
Edmonton, Canada
jag@cs.ualberta.ca

Pablo Figueroa
Department of CSE
Universidad de los Andes
Bogota, Colombia
pfiguero@uniandes.edu.co

***Abstract*—Human-in-the-loop control of remote robot arms and hands, telemanipulation, is commonly done by using a mechanical master device specifically designed to match the robot and its degrees-of-freedom (DOF). For simultaneous arm and hand manipulation, this device might be complex and expensive. In order to introduce telemanipulation into human environments, intuitive and inexpensive interfaces are needed. Most people carry smartphones, and thus smartphone-based interfaces allow people to be ”on-call” to intervene with a robot while going about their other daily activities. In this paper, we design and compare three interfaces on an inexpensive smartphone to telemanipulate a Barrett WAM arm and hand, and we use a conventional gamepad interface as a reference. Visual feedback is provided by streaming video of the robot's workspace to the smartphone. We establish the completion time, pose error and energy consumption while picking-and-placing an object in a particular location as objective measurements. We combine these measurements with a Likert-type qualitative evaluation. Results show that smartphone-based interfaces are a good compromise between availability and performance in human-in-the-loop scenarios. Nevertheless, conventional gamepad interfaces are still at least 43% better in regard to completion time and 29% in relation to pose error, but they are 24% less efficient concerning energy consumption.**

***Keywords*-teleoperation; human-robot interface; smartphone**

I. INTRODUCTION

Teleoperation of robot arms and hands have applications in diverse areas such as robotics, space and medicine. The recently completed DARPA challenge [1] simulated rescue robots in an unstructured disaster scene. Teams of 2-4 humans teleoperated a robot to perform rescue tasks, and these operators were assigned to control different sub-parts of the robot. An analysis of the competition performance showed that the performance of various sub-tasks depended on the design of the human-robot interface (HRI) to map human commands to robot motions. In particular, mappings that allowed humans to intuitively control a task performed better than those that allowed them to control individual arms and legs DOF [2]. Conventional telemanipulation uses mechanical master devices, and these devices range from joysticks to high-DOF exoskeletons. More recently mechanical devices have been replaced with screen interfaces. Ciocarlie et al. [3] presented a system that allows a motor-impaired person to command a multipurpose robot (PR2) in order to complete household tasks. In telemanipulation mode, a rendering of the robot was shown on the screen, and users operated sliders to manipulate each joint. In a similar vein, Hashimoto et al. [4] presented the TouchMe system, which allows users to control a multi-DOF robot by directly touching it on a desktop touchscreen. The touchscreen showed a view of the world from a third-person perspective. Other works seek to reduce the complexity for the human by replacing direct robot control by pointing to external objects. For example, Perez et al. [5] introduced a vision-based interface to help upper-body disabled people perform everyday tasks whilst reducing robot instruction from a 6-DOF task to either a 2-DOF pointing task or a 2D image selection task. Along the lines of the aforementioned papers, we study interfaces for human-in-the-loop interaction but, instead of using a desktop and custom-made devices, we explore interfaces based on smartphones (Figure 1).

Figure 1. User teleoperating a WAM robot arm and hand through a smartphone.

Smartphones have become widely popular over the last decade. Statista estimates that the number of users has surpassed 2 billion in 2016 [6]. These devices have a high enough computing capacity for video processing, and have broad variety of inputs and sensors. Smartphones have been exploited by researchers and used as an interaction interface between a humans and robots. Some examples are an underwater vehicle used to monitor harmful aquatic processes [7], a smartphone-based robotic platform for indoor positioning [8] and a micro-unmanned aerial vehicle [9]. In terms of manipulation tasks, Mast et al. [10] focus on the assistance

978-1-5386-3057-0/17/$31.00 ©2017 IEEE

of elderly people performing physical tasks at home. They worked together with remote human operators (informal caregivers and professional teleassistants) to teleoperate the robots during challenging tasks. This work is currently in the implementation stage using the Care-O-bot 3 mobile manipulator platform, and have concluded so far that smartphones are ideal for informal caregivers, e.g. family members or close friends.

In this paper, we compare three different smartphone-based interfaces for the teleoperation of a remote robot arm. Although the current task is 2D, it is the first step toward the development of a remote control for a robot with more degrees of freedom –such as when a robot vehicle has an arm on top of it. We believe that smartphones will become one of the most popular communication means between humans and robots; therefore, our objective is to investigate the features of their telecontrol interfaces in detail and set some guidelines for the design of smartphone interfaces that would enable human-robot interaction through telemanipulation.

The remaining sections are organized as follows. Section II describes our overall system. Then, Section III details the implementation of the proposed system. In Section IV, we present the user study assessment and the corresponding analysis in Section V. Finally, we present our conclusions in Section VII and the future work for the interfaces described throughout the paper in Section VIII.

II. System description

Our system consists of a robot arm (7-DOF WAM Arm and 4-DOF Barrett Hand), a ROS [11] master computer, a wireless switch, and a user with a smartphone, as shown in Figure 2.

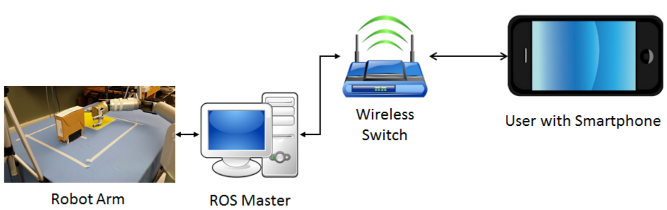

Figure 2. Hardware used in the telemanipulation system

Using the four proposed interfaces, users are able to control the position, orientation, and grasp of the end-effector. Initially, the user was located in the same building as the robot arm, so that we could focus on interaction styles rather than worrying about communication latency concerns (latency was less than 80ms). When the interfaces receive an input from the user, they send the data of three basic operations to the ROS Master: **position**, **orientation** and **grasp**. The ROS Master processes the data sent by the interfaces and generates the execution commands for the robot arm. The execution commands use a direct control strategy. In all interfaces, a top-down view of the robot arm workspace is streamed to the smartphone screen. The only difference between the interfaces is the way a user input is acquired.

III. Implementation

Given that the communication system of the robot arm works in ROS, we implemented the interfaces using ROS-Android in order to communicate the smartphone and the robot arm system. The following subsections present the interfaces we are comparing.

A. Gesture Interface

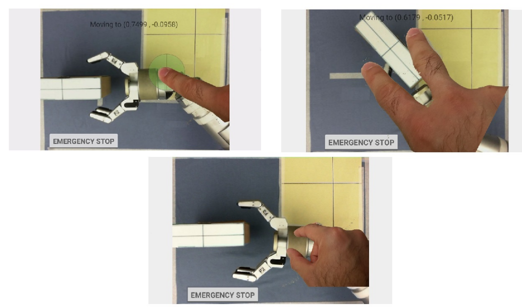

Figure 3. Gesture Interface. Upper-left picture shows drag gesture, upper-right picture shows rotation gesture, and down picture shows grasp gesture

This interface uses direct manipulation gestures that users do on the touchscreen. Gesture detection is available on most smartphones, and most smartphone users find this type of interaction quite familiar. Video feedback shows the workspace to the user so that he can control the movements of the robot arm by situating his fingers over the screen. This way, the subject can relate what he sees to what he touches and manipulates on the screen (Figure 3). In this interface, we tested several touchscreen gestures and decided to use gestures that were already known to smartphone users and which felt natural and intuitive when controlling the three basic operations.

- The **position** of the end-effector is controlled by a drag gesture. The user can 'drag' the wrist of the end-effector on the screen.
- The **orientation** of the end-effector is controlled by a rotation gesture. The change of the orientation of the end-effector is proportional to the change of the rotation gesture.
- The **grasp** of the end-effector is controlled by a pinch gesture. The grasp of the end-effector is proportional to the gap of the pinch gesture. However, the user can make a fast-pinch gesture to completely open or close the end-effector.

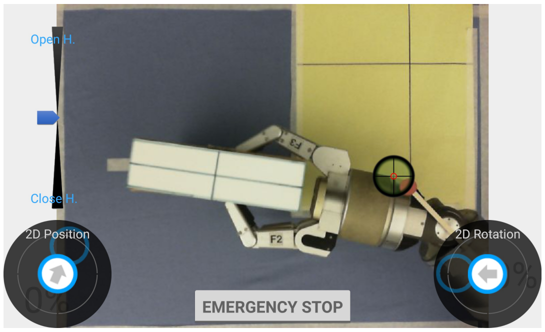

Figure 4. On-screen joysticks Interface

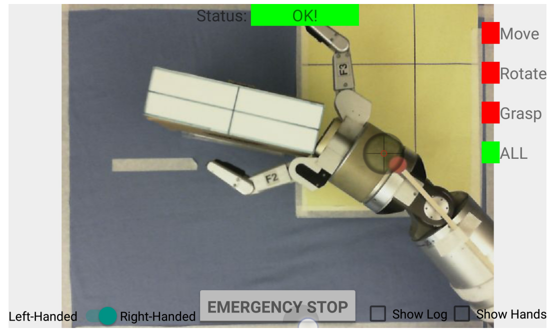

Figure 5. Leap Motion Interface

B. On-Screen Joysticks Interface

This interface draws three controls on the screen: Two 2-DOF touchscreen sticks and a 1-DOF slider (Figure 4). The sticks are very similar to those found in gamepads because such a design is familiar to users that have played games in their smartphones before.

- The joystick at the bottom left-hand corner of the screen controls the **position** of the end-effector. The velocity of displacement of the end-effector is proportional to this joystick's vertical and horizontal movement.
- The Joystick at the bottom right-hand corner of the screen controls the **orientation** of the end-effector. The change of the orientation of the end-effector is proportional to this joystick's horizontal movement.
- The slider at the left side of the screen controls the **grasp** of the end-effector. The grasping speed is proportional to the slider values. However, edge values on the slider allow the user to completely open or close the end-effector.

We positioned the joysticks at the lower corners of the screen, so that they were closer to the user's thumbs. By placing the joysticks in opposite positions, users were able to control the position (left thumb) and orientation (right thumb) of the end-effector at the same time. Initial tests showed that users often tried to fix the orientation while trying to grasp an object, so the slider was positioned at the left side of the screen to enable users to simultaneously perform both actions.

C. Leap Motion Interface

This interface enables the connection of a Leap Motion device to our system. Given that the task involves controlling a robot hand, using the direct movements of the user's hand might be more natural for them. The interface tracks gestures on the left hand to enable or disable the movements of the robot hand, and it also tracks the joints of the user's right hand to control the robot arm (Figure 5). In order to achieve a natural interaction when controlling the robot hand, we decided to make a direct mapping between the robot hand and the user's hand.

- The position of the user's wrist controls the **position** of the end-effector. The displacement of the end-effector is proportional to the user's wrist displacement.
- The orientation of the user's hand controls the **orientation** of the end-effector. The change of the orientation of the end-effector is proportional to the user's hand yaw rotation.
- The grasp of the user's hand controls the **grasp** of the end-effector. The grasp of the end-effector is proportional to the user's hand grasp. The user can fully open or close the end-effector by completely opening or closeing his own hand.

D. Gamepad Interface

Figure 6. Gamepad Interface

This interface was created as a reference, in order to compare its performance with the previous three smartphone interfaces. It enables the connection of a standard gamepad in our system. The interface displays the default gamepad scheme on the screen, as shown in Figure 6.

- The left joystick on the gamepad controls the **position** of the end-effector. The velocity of displacement of the end-effector is proportional to this joystick's vertical and horizontal movement.
- The right joystick on the gamepad controls the **orientation** of the end-effector. The change of the orientation of the end-effector is proportional to this joystick's horizontal movement.

- The triggers on the back of the gamepad control the **grasp** of the end-effector. The left trigger closes the end-effector, and the right trigger opens it. The grasp speed of the end-effector is proportional to how hard the user presses the triggers, given that the triggers have 16 levels of sensibility. However, the the user can fully open or close the end-effector by completely pressing the corresponding trigger.

Our gamepad scheme resembles that of most shooter videogames where a character's position is controlled using the left stick while its rotation is controlled using the right stick. For this reason, we decided to map the position and orientation of the end-effector to the sticks the same way. For the grasp of the end-effector, we needed another continuous input -not a digital button-, so the triggers were an evident choice; 'right' to grasp the object and 'left' to release it.

IV. User Study

A user study was conducted with eight volunteers and lasted an average of 40 minutes per user. A small number of users is relatively common in robotics evaluations [12], [3], [13], [14], [15], where setup-time and experiment time is typically much longer than in a human-computer interface (HCI) or psychology experiment. A short practice session was conducted before the test. The order of the interfaces was counter balanced between subjects, and participants performed a within-subject evaluation task using each interface. We video-recorded the robot arm while the users were completing the tasks in order to get the objective measurements. Finally, users rated the interfaces through a Likert-type qualitative evaluation.

A. User profile

Subjects were between 25 and 34 years old. There were 7 male and 1 female participants. One male was left-handed. We asked the users to give a subjective punctuation (from 1 to 5) to their level of experience using a touchscreen, a gamepad, and a Leap Motion. Figure 7 shows that users had the best level of experience using a touchscreen (4.5), a good level experience using a gamepad (3.87),and a lousy level of experience using a Leap Motion (1.25). Three subjects had normal vision, the remaining five had their vision corrected to normal.

B. Practice Exercise

Before collecting data, each user had a couple of minutes to practice using each interface to move an object from one place to another. After a user finished practicing with an interface, we asked if he had any doubts regarding how he was supposed to use the interface. If everything was clear, he could move on to practice with another interface.

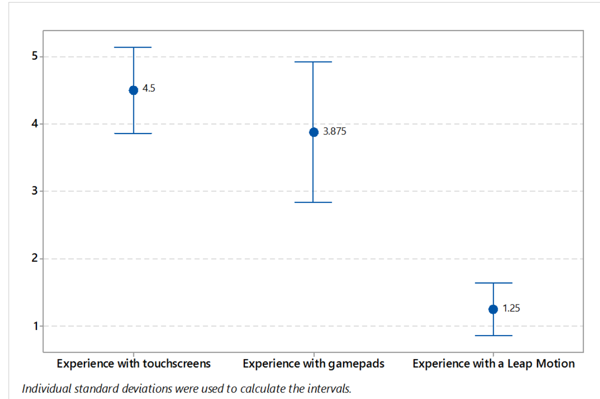

Figure 7. Users' level of experience using a touchscreen, a gamepad and a Leap Motion

C. Assessment task

The assessment involved teleoperating a robot arm to pick-and-place an object on a target at a specific location and with a desired alignment. The object was grasped on the left side of the screen and had to be placed on top of the yellow target in such a way that the vertical and horizontal cross-hair lines of the target location matched the lines on top of the object (Figure 8). We asked the users to complete the task in a short time and align the object and the target as best as possible.

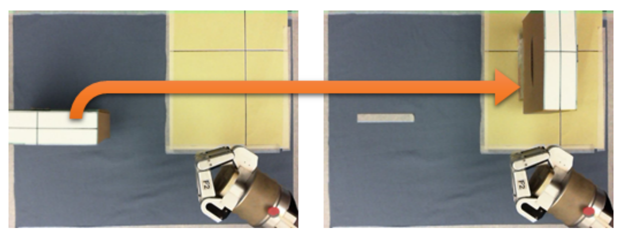

Figure 8. Task the users had to complete.

V. Experimental Results

The following subsections present the data measured throughout the user study. We recorded videos of the workspace and stored the arm's pose while each task was being completed. At the end of the session we carried out a brief user survey.

A. Objective Measurements

We recorded the time until completion, pose error and energy consumption for each task. Figure 9 shows each interface's time performance. The average time until completion when using the gesture interface was 110.23s, while the corresponding time for the on-screen joystick interface was 108.65s, thus making the latter somewhat faster than the gesture interface. It is important to mention that the gesture interface had an outlier value (Figure 9), and this situation

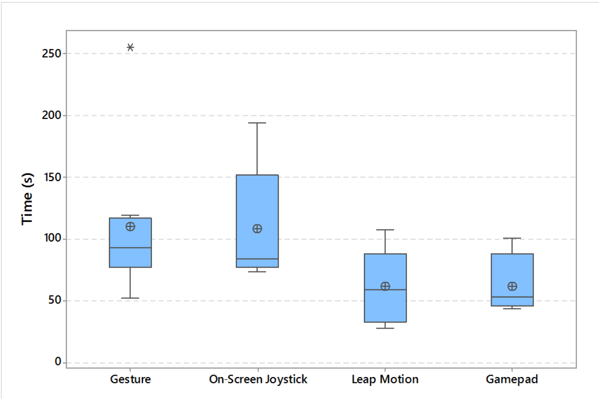

Figure 9. Time taken to complete the task for each interface. Symbols $*$ and $\oplus$ represent outlier and mean values respectively.

resulted in an increase of the average time of the interface so that it became the longest of all four. The time when using the Leap Motion interface was 61.99s, which made it 42.94% faster than the on-screen joystick interface. This difference is significant with a statistical confidence level of 95%. The average time when using the gamepad interface was 62.09s which is somewhat slower than the Leap Motion interface. However, this difference is not significant with a statistical confidence level of 95%. The results show that the gamepad interface is the fastest interface for this task although the time difference with respect to the Leap Motion interface is insignificant. It is interesting to note that touchscreen-based interfaces (gestures and on-screen joysticks) had a similar average time (around 110s).

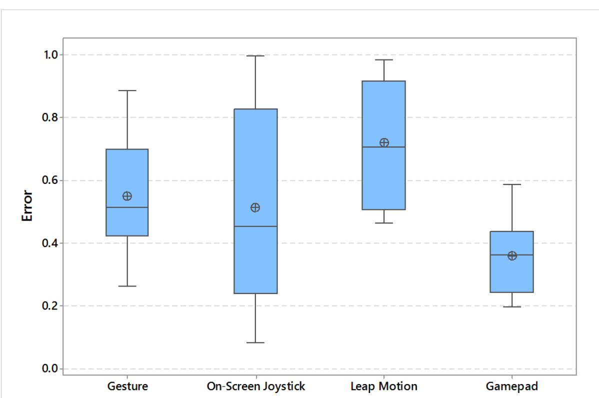

Figure 10. Normalized error when positioning the object over the target for each interface. The symbol $\oplus$ represents mean values.

Figure 10 shows the value of the error when positioning the object on the target. These errors were calculated by measuring the distance between the object's and target's centers respectively, and the difference between the object's and the target's yaw orientation. These two errors were averaged and normalized from 0 to 1. The average pose error when using the Leap Motion interface was 0.72, while the corresponding number for the gesture interface infringements was 0.55 (23.61% more precise than the Leap Motion). The pose error of the gamepad interface was 0.36, while the pose error of the on-screen joystick interface was 0.51. The results in Figure 10 show that the gamepad interface is also the most precise interface for this task, and its performance when compared to the gesture and Leap Motion interfaces is significant with a statistical confidence level of 95%. We produced a scatter graph in order to check if there are any relationships between the error and the time values for each interface. Figure 11 illustrates the normalized error versus the normalized time for each interface. The outlier from the gesture interface was removed in order to reduce the range of time covered in the figure.

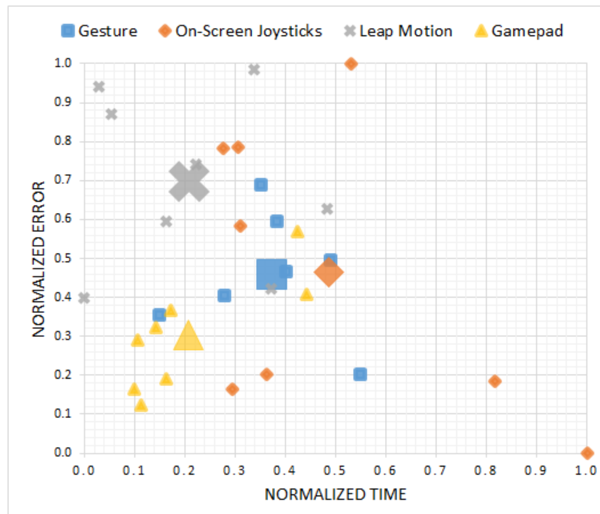

Figure 11. Error versus time scatterplot. Small icons are individual users and large icons represent the interfaces' mean.

The Leap Motion interface's scatter is closer to the top left corner of the graph. Thus the interface fast but error-prone with respect to positioning. The gamepad interface's scatter is closer to the bottom left corner of the graph, which shows that this interface is both fast and less prone to positioning errors. The gesture interface's scatter is close to the middle-left side of the graph; therefore, it is slower than the gamepad interface while still error-prone. The gesture interface's scatter is also less susceptible to positioning errors than the Leap Motion interface, but it is slower than the latter. The on-screen joysticks interface's scatter has a high range for both the error and time; however, its average error and time are somewhat better than those of the gesture interface.

By recording the robot arm's pose when performing the task, we discovered that its acceleration is different for each interface. Figure 12 shows that the on-screen joystick interface produces the lowest acceleration changes for the

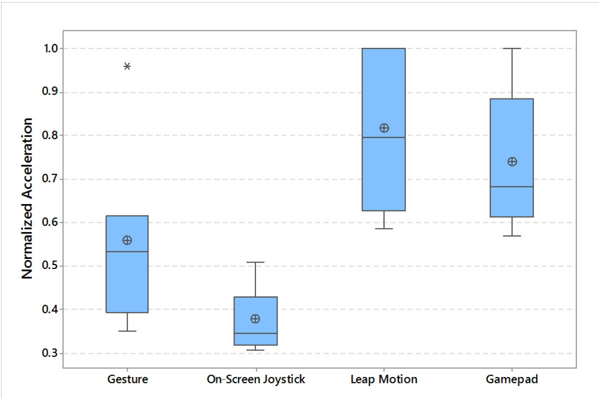

Figure 12. Normalized total changes in acceleration. Symbols $*$ and $\oplus$ represent outlier and mean values respectively.

robot arm followed by the gesture interface and the gamepad interface. The Leap Motion interface produces the biggest acceleration changes of all interfaces thus causing the robot arm to require more energy.

B. Subjective Measurements

In order to evaluate the interfaces, the users were asked to fill out a Likert-type survey after the test had finished. We asked the following questions:

- Completing the task using the interface was: 1 (Hard) ... 5 (Easy)
- The sensation of naturalness of the interface was: 1 (Unnatural) ... 5 (Natural)
- If you had to use this app on a daily basis, arrange the interfaces from the one you would use the most to the one you would use the least.
- How physically demanding was the Interface? 1 (Not demanding) ... 5 (Demanding)
- How mentally demanding was the Interface? 1 (Not demanding) ... 5 (Demanding)

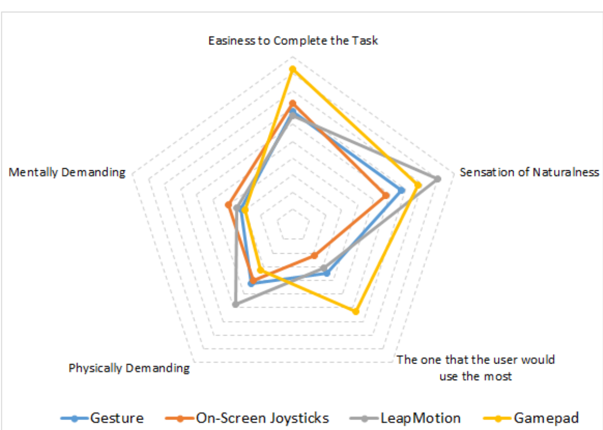

Figure 13. Scores comparing the different interfaces used.

Regarding the ease to complete the task, the results in Figure 13 show that the on-screen joystick interface was slightly easier for most participants than the gesture and Leap Motion interfaces. The Leap Motion interface; however, feels more natural among users than the gesture and on-screen joystick interfaces although Leap Motion interface produces more physical stress. Moreover, gesture interface is preferred when compared to the Leap Motion and on-screen joystick interfaces, even though it produces more mental stress on users when completing the task. Overall, the gamepad interface has proven to be the preferred interface, and according to the participants, it is the easiest interface. Figure 13 shows that the gamepad interface's score has the best values for 4 out of 5 questions.

VI. DISCUSSION

The conventional gamepad interface had the best performance in our study and remains the favorite among most users. However, we noted certain behaviors when analysing the data from the other three interfaces. For example, if the objective were to complete the task faster, the Leap Motion interface would be a great choice. However, if precision is the goal, the gesture and on-screen interfaces would be a better choice. Overall, gesture and on-screen joystick interfaces are less stressing for users, so they could be used for longer periods of time than the Leap Motion interface. It is interesting to note that the standard deviations for the on-screen interface in Figure 9 and 10 is considerably bigger than for the other interfaces. This behavior means that some users completed the task with low pose error when using this interface, but it also means that other subjects had high pose error when using this interface. More training may help to regularize this behavior.

VII. CONCLUSIONS

In this article, we presented three smartphone-based interfaces to operate a robot arm in a human-in-the-loop scenario. A visual feedback of the scenario was given to the user through a top-down view. The performance of such interfaces was assessed through a user study. The human-in-the-loop scenario can be successfully carried out with the current smartphone-based interfaces; nonetheless, the results show that the conventional gamepad interface is still a very challenging reference to overmatch. Issues remain in the interfaces' level of control as we are using a low level of control -direct control-. For this reason, a higher level of autonomy will be considered next user studies.

Our system was implemented using a Barrett's WAM arm and hand, and a smartphone. The user study was carried out by eight users. We found that the reference interface (gamepad) had the best time and precision. The gesture interface followed in precision but with longer task times because users could only control one robot movement (one gesture) at a time. The Leap Motion interface followed with quick task completion times but with poor precision. A challenge with the Leap Motion interface is for the user to hold and move his arm and hand precisely without any physical feedback like the joysticks in a gamepad, or a surface to rest, like the touchscreen, thus making it more demanding for them. We expected the on-screen joysticks interface to have similar results as the gamepad interface, but its performance was not as good as expected. We believe that this outcome is due to the lack of physical feedback from the interface.

Users felt that the task was easiest to complete when using the gamepad interface, an interface they had previous experience with. The on-screen joysticks interface followed and then the gesture interface. The Leap Motion interface was last, an interface they had little to none previous experience with. Subjects felt that the sensation of naturalness was better using the Leap Motion interface; we believe this is a result of the DOF mapping being more intuitive than that the other interfaces. The gamepad interface followed and then the gesture interface. The on-screen joysticks interface was last although its grade was not so low. The gamepad interface seems to be the favorite among the users, probably because of its ease of use, followed by the gesture interface and the Leap Motion interface.

Overall, it is perhaps not surprising that the physical joysticks in the gamepad, that were specifically designed to control motion, perform better than on-screen joysticks implemented in software on a smartphone. The direct manipulation touchscreen, gesture interface, had a slightly better mean than the on-screen joysticks but, perhaps more important, less variance in error and time, allowing more consistent task performance. One inherent challenge with a touchscreen is that the physical friction force is highest at zero speed (sticktion), then goes down as the user moves faster. This is the opposite of what is desired. A HRI interface should provide feedback that is proportionally stronger with the users control action –like the springs does in a gamepad. Despite this, a touchscreen interface is sufficiently usable, and a desirable interface to give humans an "anytime" interface to operate robots through the smartphone that they ordinarily carry anyway. As robots take on new tasks in unstructured environments, they will encounter situations they were not programmed for. Then, a smartphone interface will enable humans to take direct control of the robot without being physically present. One use of would be a robot monitoring and performing routine tasks at a remotely located utility infrastructure such as electric power substations, pipeline pumping stations, well-heads, etc. In an emergency, the robot would ring the human, and the human can use the robot camera to assess the damage and take appropriate mitigating action, e.g. shutting down the faulty parts and recovering use of the remaining part.

VIII. FUTURE WORK

The gamepad interface (our reference) was clearly the most accurate, fast and favorite interface for most users. These days most people are familiar with joystick control, and our choice to map the joystick motions in a similar way to a first person shooter game likely helped users. We believe we can improve the other smartphone interfaces so that they can get closer to, or even better than, the gamepad interface. We heard the users' feedback and suggestions, and we defined some changes that may improve the interfaces.

For the direct manipulation gesture interface, we want to implement stateless gestures, where users can simultaneously control the position, orientation and grasp of the robot hand by using two fingers. This way, users are not forced to lift their fingers off the screen (they still will be able to do it if they want to) to change the gesture they are doing. A screen-gesture interface would likely have a higher relative advantage compared to a gamepad if a large number of DOFs are controlled, e.g. all fingers of a 5-finger robot hand.

For the on-screen joystick interface, we want to change the slider for a rolling selector, so that users do not have to keep touching the slider until the desired grasp is reached. We also plan to implement movable joysticks because we noticed users were exceeding joystick surfaces when interacting with them. With movable joysticks, users are not restricted to a fixed area on the screen, but they are free to move their fingers across the screen while still being able to reach the joysticks.

The Leap Motion interface was too sensitive to the movements of the users' hand and fingers. As a result, users became physically and mentally stressed. We need to ease this stress and increase the precision of the interface. The Leap Motion has also an additional delay while processing the captured images, which impares the users coupling to the robot motion.

Finally, we need to address greater delays in a remote teleoperation environment, where the delay is significant. This can be addressed in motor coordinates using passivity, or visually using predictive display[13]. Also, other feedback such as audio and/or motion/vibration feedback could be included to allow users to be more aware of the robot's state.

REFERENCES

[1] DARPA, "DARPA Robotics Challenge," http://www.darpa.mil/program/darpa-robotics-challenge, 2016, [Online; accessed 15-Nov-2016].

[2] H. A. Yanco, A. Norton, W. Ober, D. Shane, A. Skinner, and J. Vice, "Analysis of human-robot interaction at the darpa robotics challenge trials," *Journal of Field Robotics*, vol. 32, no. 3, pp. 420–444, 2015. [Online]. Available: http://dx.doi.org/10.1002/rob.21568

[3] M. Ciocarlie, K. Hsiao, A. Leeper, and D. Gossow, "Mobile manipulation through an assistive home robot," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 5313–5320.

[4] S. Hashimoto, A. Ishida, M. Inami, and T. Igarashi, "Touchme: An augmented reality based remote robot manipulation," in *The 21st International Conference on Artificial Reality and Telexistence, Proceedings of ICAT2011*, 2011.

[5] C. Perez Quintero, O. Ramirez, and M. Jagersand, "Vibi: Assistive vision-based interface for robot manipulation," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4458–4463.

[6] Statista, "Number of smartphone users worldwide from 2014 to 2020," https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/, 2016, [Online; accessed 02-Jan-2017].

[7] Y. Wang, R. Tan, G. Xing, J. Wang, X. Tan, and X. Liu, "Samba: a smartphone-based robot system for energy-efficient aquatic environment monitoring," in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*. ACM, 2015, pp. 262–273.

[8] W.-W. Kao and B. Q. Huy, "Indoor navigation with smartphone-based visual slam and bluetooth-connected wheel-robot," in *Automatic Control Conference (CACS), 2013 CACS International*. IEEE, 2013, pp. 395–400.

[9] B. Zhao, T. Hu, D. Zhang, L. Shen, Z. Ma, and W. Kong, "2d monocular visual odometry using mobile-phone sensors," in *Control Conference (CCC), 2015 34th Chinese*. IEEE, 2015, pp. 5919–5924.

[10] M. Mast, Z. Materna, M. Španěl, F. Weisshardt, G. Arbeiter, M. Burmester, P. Smrž, and B. Graf, "Semi-autonomous domestic service robots: Evaluation of a user interface for remote manipulation and navigation with focus on effects of stereoscopic display," *International Journal of Social Robotics*, vol. 7, no. 2, pp. 183–202, 2015. [Online]. Available: http://dx.doi.org/10.1007/s12369-014-0266-7

[11] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," *ICRA workshop on open source software*, vol. 3, no. 3.2, p. 5, 2009.

[12] D. Sakamoto, K. Honda, M. Inami, and T. Igarashi, "Sketch and run: A stroke-based interface for home robots," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '09. New York, NY, USA: ACM, 2009, pp. 197–200. [Online]. Available: http://doi.acm.org/10.1145/1518701.1518733

[13] D. Lovi, N. Birkbeck, A. H. Herdocia, A. Rachmielowski, M. Jgersand, and D. Cobza, "Predictive display for mobile manipulators in unknown environments using online vision-based monocular modeling and localization," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2010, pp. 5792–5798.

[14] K. M. Tsui, J. M. Dalphond, D. J. Brooks, M. S. Medvedev, E. McCann, J. Allspaw, D. Kontak, and H. A. Yanco, "Accessible human-robot interaction for telepresence robots: A case study," *Paladyn*, vol. 6, no. 1, 1 2015.

[15] J. Bohren, C. Paxton, R. Howarth, G. D. Hager, and L. L. Whitcomb, "Semi-autonomous telerobotic assembly over high-latency networks," in *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2016, pp. 149–156.