AR Indicators for Visually Debugging Robots

Bryce Ikeda
Department of Computer Science
University of Colorado Boulder
Boulder, Colorado
bryce.ikeda@colorado.edu

***Abstract*—Programming robots is a challenging task exacerbated by software bugs, faulty hardware, and environmental factors. When coding issues arise, traditional debugging techniques are not always useful for roboticists. Robots often have an array of sensors that output complex data, which can be difficult to decipher as raw text. Augmented reality (AR) provides a unique medium for conveying data to the user by displaying information directly in the scene as their corresponding visual definition. In my research, I am exploring various design approaches towards AR visualizations for expert robotics debugging support. From my initial work, I developed design guidelines to inform two future bodies of work which investigate better ways of visualizing robot sensor and state data for debugging.**

***Index Terms*—Augmented Reality (AR); Mixed Reality (MR); ARHMD; interface design; robots; debugging; HRI; HCI**

I. INTRODUCTION

Debugging in its most basic terms is defined as "the attempt to pinpoint and fix the source of an error" [1]. In addition to standard debugging challenges faced by any computer programmer (such as syntax, logic, compilation, or runtime errors) roboticists often confront complications caused by environmental factors and unreliable hardware. Debuggers such as GNU Debugger (gdb) or Python Standard Debugger (pdb) are difficult to use because breakpoints will alter the robot's behavior during the live execution of a user's code. Other debugging techniques, such as reading raw data from print statements or log files, can be tedious and confusing. For example, end effector transformations can be difficult to visualize and validate via matrices, but when inspected through 3D visualizations, can be easily confirmed.

One widely used tool for robotics programming across both academic research and industry settings is the Robot Operating System (ROS). RViz, a central component of the ROS ecosystem, enables roboticists to visualize 3D data on a 2D screen (i.e., perspective 3D [2]) [3]. Examples of data visualized in robotics include point clouds, object affordances, and robot models [4]–[6]. Although commonly used, there has been little research to evaluate the design choices of this tool. For example, RViz has a clear design flaw in its use of rainbow color schemes for data encoding [7], [8], but is still accepted as a standard debugging tool in the robotics community.

In addition to RViz, 3D immersive augmented reality (AR) tools are emerging as a third modality for visualizing robot data [9]–[17]. A key attribute separating AR interfaces is its ability to combine real and virtual objects directly in the user's physical environment, rather than on a 2D monitor [18]. This setup provides stereoscopic depth cues that have been found to improve a user's performance on tasks related to spatial understanding [19]. By projecting real-time data into the user's physical world, AR can provide an intuitive way for a roboticist to understand a robot's sensors, internal data, and operation, consequently debugging errors.

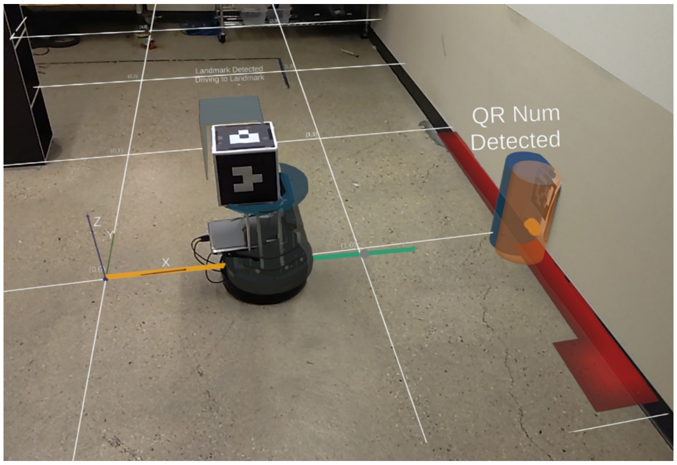

Fig. 1: My AR visual debugging tool displaying the robot's sensor and state information for a QR code detection and robot navigation task.

I am currently a second-year PhD student working in the area of Virtual, Augmented, and Mixed-Reality for Human-Robot Interaction (HRI). I am hoping to attend HRI Pioneers to hear feedback on my research from a diverse community of my peers, while also learning about other areas of exploration within the field of HRI. Additionally, as I develop AR debugging tools for roboticists, it would be helpful to discuss new ideas with those who would be directly impacted by my work.

II. RELATED WORK

Augmented reality can be an effective tool to convey robot sensor, state and task data to the user. This has been illustrated through the exploration of visualizing robot state and intent [9]–[13], [20], [21]. These works demonstrate the value of displaying digital twins and spatial data to convey important information to the user. However, there has been limited work specifically focusing on visualization design choices supporting roboticists during the debugging process. Some early systems have been developed to help users monitor the inner state and behavior of mobile robots, manipulators, and robotic swarms [20], [22]–[24]. But, these applications apply perspective 3D, which remove the depth cue of stereopsis and potentially detaches the data from its true context. Furthermore, these systems continuously display all of the information at once, possibly overwhelming the user's ability to understanding what is being displayed. To improve these systems, my work focuses on the research question *"How should we design AR debugging tools for the purpose of supporting robot programmers during the debugging process?"*

III. PRIOR WORK

My first study sought to investigate existing and emerging visualizations tools that support robotics debugging [25]. I selected a 2D Graphical User Interface (2D GUI) and RViz as examples of widely-used existing approaches and developed a 3D immersive AR debugging platform as an emerging designs (see Figure 1). I conducted a qualitative expert evaluations with 24 roboticists across two universities as a 3×1 between-participants design (each roboticist participant used one of the three visualization aids: RViz, 2D GUI, or AR). Participants were tasked with programming a mobile robot to complete two tasks that represent standard applications robot developers typically code: a *detection* task and a *finder* task [22]. Both programs require object detection and navigation and thus force the programmer to use multiple variables and control sequences. After the second task, a survey was administered to understand the participant's experiences followed by open-ended verbal feedback regarding their assigned debugging tool.

From the participant's combined subjective feedback, I performed a thematic analysis identifying three design guidelines to inform future iterations of visual debugging tools. First, a programmer's sensemaking process can be aided by intuitive visualizations that properly encode data. Visualizations such as waypoints and spatially recognizable objects, when correctly located in the environment, were intuitive to the participant's understanding of the data. This provided a clearer connection between the participant's code and the robot's behavior, thereby improving the participant's thought process during debugging. Second, by minimizing visual clutter users have an easier time identifying and synthesizing important information. While participants appreciated simple representations of data, some found that the path history occluded future path trajectories hindering their understanding of the robot's future pose. Finally, debugging interfaces should be designed to cohesively integrate information from disparate data streams with the goal of reducing cognitive load. Participants noted that combining data visualizations within a single context was helpful, but other information worth incorporating could be bug specific indicators such as topic data, code errors or warnings.

IV. FUTURE WORK

I will extend my prior work in two ways. First, our previous study indicates that sensemaking can be aided by intuitive visualizations that properly encode data. To expand on the sensemaking process, I will investigate how we can draw a user's attention to task specific debugging information by manipulating *visual saliency*. Visual saliency is the perceptual quality coined by the Visualization (VIS) community denoting areas in a scene that draw a viewer's attention such as color, luminance and edges [26]. Drawing on this concept, I pose the research question *"How can varying the appearance of data using feature and attention-based saliency techniques aid a user's ability to pinpoint program bugs?"* To answer this, I am currently prototyping different techniques for altering the saliency of robot data visualizations. These prototypes include motion (an object's digital twin moving in place) or luminance (a brighter or pulsating digital twin of the object). I will remove these indicators when the user's gaze intersects with the object (attention-based saliency). Similarly, data representations that users consistently refer to can be rendered opaque when the user's attention is directed towards it and slightly transparent when in the peripherals, thus non-distracting. To evaluate these concepts I will have participants debug a flawed robot system performing a task in which the robot sorts objects of similar shapes and sizes into their corresponding bins. The involved metrics will include bug identification efficiency and the debugging tool's effect on a user's cognitive load as well as a subjective questionnaire about their experience.

My second proposed study will look into my third design guideline; how to cohesively integrate information from disparate data streams. In the context of debugging robots, faults can occur in both the hardware and the software components. Prior work has examined the effectiveness of highlighting faulty hardware components to reduce operator anxiety during collaboration scenarios [27]. However, in this application the operator must still switch perspectives to a text box on their computer making it difficult to contextualise a problem in real time. Another typical debugging technique is to monitor relevant topic data streams leading up to a robot error. For this second study, I pose the research question *"How can we superimpose topic data, warnings and errors into locations that promote the pinpointing of robot errors?"* To answer this, I will conduct an online questionnaire to understand where programmers prefer to see this information within the context of their robot work space. We will show the participants an interactive view of a robotic system and ask them to place topic data, error messages and warnings where it is most intuitive to the user. After, I will implement the results on an AR system and compare it against a 2D monitor baseline with roboticists programming and debugging a physical robot.

V. CONCLUSION

The most successful robot systems are the ones with the fewest amount of bugs. Current systems have failed to keep up with the needs of roboticists and must evolve with robots themselves. My goal is to provide a more intuitive way of analyzing robot data for both programming and debugging purposes. In doing so, roboticists will be able to create higher quality applications in less time.

REFERENCES

[1] Michael Kölling David J. Barnes. *Objects First with Java: A Practical Introduction Using BlueJ 6*. Objects First with Java: A Practical Introduction Using BlueJ 6th Edition. Pearson, 6th edition, 2016.

[2] John P. McIntire, Paul R. Havig, and Eric E. Geiselman. Stereoscopic 3d displays and human performance: A comprehensive review. *Displays*, 35(1):18–26, 2014.

[3] Hyeong Ryeol Kam, Sung-Ho Lee, Taejung Park, and Chang-Hun Kim. Rviz: a toolkit for real domain data visualization. *Telecommunication Systems*, 60(2):337–345, 2015.

[4] Stephen Hart, Paul Dinh, and Kimberly A Hambuchen. Affordance templates for shared robot control. In *2014 AAAI Fall Symposium Series*, 2014.

[5] Sebastian Pütz, Thomas Wiemann, and Joachim Hertzberg. Tools for visualizing, annotating and storing triangle meshes in ros and rviz. In *2019 European Conference on Mobile Robots (ECMR)*, pages 1–6. IEEE, 2019.

[6] D Chikurtev, I Rangelov, N Chivarov, E Markov, and K Yovchev. Control of robotic arm manipulator using ros. *Bulgarian Academy of Sciences-Problems of Engineering Cybernetics and Robotics*, 69:52–61, 2018.

[7] David Borland and Russell M Taylor II. Rainbow color map (still) considered harmful. *IEEE Computer Architecture Letters*, 27(02):14–17, 2007.

[8] Danielle Albers Szafir. The good, the bad, and the biased: Five ways visualizations can mislead (and how to fix them). *Interactions*, 25(4):26–33, 2018.

[9] Hooman Hedayati, Michael Walker, and Daniel Szafir. Improving collocated robot teleoperation with augmented reality. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '18, page 78–86, New York, NY, USA, 2018. Association for Computing Machinery.

[10] Eric Rosen, David Whitney, Elizabeth Phillips, Gary Chien, James Tompkin, George Konidaris, and Stefanie Tellex. *Communicating Robot Arm Motion Intent Through Mixed Reality Head-Mounted Displays*, pages 301–316. 01 2020.

[11] Michael Walker, Hooman Hedayati, Jennifer Lee, and Daniel Szafir. Communicating robot motion intent with augmented reality. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '18, page 316–324, New York, NY, USA, 2018. Association for Computing Machinery.

[12] Michael E Walker, Hooman Hedayati, and Daniel Szafir. Robot teleoperation with augmented reality virtual surrogates. In *2019 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2019.

[13] Christopher Reardon, Kevin Lee, John G Rogers, and Jonathan Fink. Augmented reality for human-robot teaming in field environments. In *International Conference on Human-Computer Interaction*, pages 79–92. Springer, 2019.

[14] Matthew B Luebbers, Connor Brooks, Minjae John Kim, Daniel Szafir, and Bradley Hayes. Augmented reality interface for constrained learning from demonstration. In *Proceedings of the International Workshop on Virtual, Augmented, and Mixed Reality for HRI (VAM-HRI), Daegu, Korea (South)*, pages 11–14, 2019.

[15] Connor Brooks and Daniel Szafir. Visualization of intended assistance for acceptance of shared control. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11425–11430, 2020.

[16] Andre Cleaver, Muhammad Faizan, Hassan Amel, Elaine Short, and Jivko Sinapov. SENSAR: A visual tool for intelligent robots for collaborative human-robot interaction. *CoRR*, abs/2011.04515, 2020.

[17] Zhanat Makhataeva and Huseyin Atakan Varol. Augmented reality for robotics: a review. *Robotics*, 9(2):21, 2020.

[18] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6):34–47, 2001.

[19] John P McIntire, Paul R Havig, and Eric E Geiselman. What is 3d good for? a review of human performance on stereoscopic 3d displays. In *Head-and Helmet-Mounted Displays XVII; and Display Technologies and Applications for Defense, Security, and Avionics VI*, volume 8383, page 83830X. International Society for Optics and Photonics, 2012.

[20] Tomáš Kot and Petr Novák. Utilization of the oculus rift hmd in mobile robot teleoperation. In *Applied Mechanics and Materials*, volume 555, pages 199–208. Trans Tech Publ, 2014.

[21] Kishan Chandan, Vidisha Kudalkar, Xiang Li, and Shiqi Zhang. Negotiation-based human-robot collaboration via augmented reality. *arXiv preprint arXiv:1909.11227*, 2019.

[22] Toby Collett and Bruce A. MacDonald. An augmented reality debugging system for mobile robot software engineers. *Journal of Software Engineering for Robotics*, 1:18–32, 2010.

[23] Alan G. Millard, Richard Redpath, Alistair M. Jewers, Charlotte Arndt, Russell Joyce, James A. Hilder, Liam J. McDaid, and David M. Halliday. Ardebug: An augmented reality tool for analysing and debugging swarm robotic systems. *Frontiers in Robotics and AI*, 5:87, 2018.

[24] Fabrizio Ghiringhelli, Alessandro Giusti, Jerome Guzzi, Gianni Di Caro, Vincenzo Caglioti, and Luca Maria Gambardella. Interactive augmented reality for understanding and analyzing multi-robot systems. 09 2014.

[25] Bryce Ikeda. Advancing the design of visual debugging tools for roboticists. In *2022 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2022.

[26] Laura E Matzen, Michael J Haass, Kristin M Divis, Zhiyuan Wang, and Andrew T Wilson. Data visualization saliency model: A tool for evaluating abstract data visualizations. *IEEE transactions on visualization and computer graphics*, 24(1):563–573, 2017.

[27] Giancarlo Avalle, Francesco De Pace, Claudio Fornaro, Federico Manuri, and Andrea Sanna. An augmented reality system to support fault visualization in industrial robotic tasks. *IEEE Access*, 7:132343–132359, 2019.