DISA 2018

1st World Symposium on Digital Intelligence for Systems and Machines

PROCEEDINGS

IEEE

August 23-25, 2018
Košice, Slovakia

DISA 2018

IEEE World Symposium on Digital Intelligence for Systems and Machines

PROCEEDINGS

August 23-25, 2018
Košice, Slovakia

Website: http://www.disa2018.org/

ISBN: 978-1-5386-5101-8
IEEE Catalog number: CFP18P13-USB
Proceedings editor: Ján Paralič, Peter Sinčák, Laszlo Kovacs, XenFuWang

TABLE OF CONTENTS

Invited papers

Various Aspects of Digital Intelligence

Speech, language and linguistic issues for Human Computer Interactions
(Special Session)

Implementation of a Speech Enabled Virtual Reality Training Tool for Human-Robot Interaction

Christopher R. Hudson, Cindy L. Bethel,
and Daniel W. Carruth,
Center for Advanced Vehicular Systems,
High Performance Computing Collaboratory (HPC^2),
Bagley College of Engineering,
Mississippi State University,
200 Research Blvd., Starkville, MS 39759.
Email: chudson@cavs.msstate.edu,
cbethel@cse.msstate.edu, dwc2@cavs.msstate.edu

Matus Pleva, Stanislav Ondas,
and Jozef Juhar,
Department of Electronics
and Multimedia Communications,
Faculty of Electrical Engineering and Informatics,
Technical university of Kosice,
Letna 9, 04120 Košice, Slovakia.
Email: matus.pleva@tuke.sk,
stanislav.ondas@tuke.sk, jozef.juhar@tuke.sk

***Abstract*—This paper presents the outline and implementation of a virtual reality-based training tool for interacting with the Julius Open-Source Voice Recognition Software [1]. This implementation builds on our previously designed tool for training police officers the command words associated with an integration of Julius with a robotic system with which they trained. By expanding the design into virtual reality, it was possible to increase realism of training as well as rapidly prototype and test new interfaces to support individuals' interactions with the voice recognition system. A simulated wearable display was provided to allow officers to look up the commands they needed and to provide feedback on what the voice recognition system detected. Additional functionality was incorporated to support new interaction methods such as push to talk and push to confirm.**

***Keywords*—Human-Robot Interface, Automatic Speech Recognition, Training Tool, Virtual Reality**

I. Introduction

There are many situations in which training users presents significant challenges. Training law enforcement officers in the real world can present logistical challenges of reserving locations suitable for law enforcement operations, recruiting and scheduling role players, making available dangerous or expensive equipment outside of regular work hours, etc. Training law enforcement officers to use a voice recognition system to control a robot during tactical operations presents all of these challenges.

The use of virtual reality (VR) for training is increasingly popular as VR allows individuals to train on a wide range of tasks in a realistic but safe and controlled environment. VR allows individuals to safely interact with equipment that might be dangerous or expensive, without risking damage to the equipment or the person. In addition, VR allows trainees to gain experience in scenarios that rarely occur or are impractical to recreate in the real world. VR also allows individuals to train in locations that are simply inaccessible for real-world training but are realistic environments they could expect to operate in during an emergency.

In addition to the benefits for trainees, VR can also serve as an engine for innovation. Implementing a complex system with multiple real-world components often presents numerous technical challenges (e.g., networking and communication, basic wiring, noisy environments, etc.). In VR, many of these complications can be abstracted away and allow designers to rapidly prototype interfaces to complex systems that can be tested in VR and then later translated to the real world.

II. Related Work

Using virtual reality as a tool for effective training is not a new concept. Several virtual reality training environments have been developed for a multitude of tasks including CBRN [3], first responders [2], aircraft cabin safety [4], and improvement of communication skills [5]. These tools are currently being adopted across the United States by major retail companies such as Walmart [6].

III. Previous Work

Previously, Julius [1] was integrated with the Dr. Robot Jaguar V4 robot platform in use at Mississippi State University (MSU). The Julius open source software was implemented thanks to collaborations with the Technical University of Kosice (TUKE), who configured the grammar and phonetic

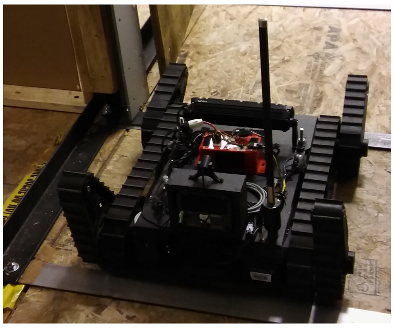

Fig. 1. Dr. Robot Jaguar V4 robot with attached distraction devices

dictionary needed and compiled the acoustic models from the Wall Street Journal (WSJ) [7], TIMIT (Texas Instruments and Massachusetts Institute of Technology) [8] and TUKE-BNews (TUKE - Broadcast News) [9] acoustical corpuses. Additionally, a network socket interface was developed for sending the recognized commands to the Robot Operating System (ROS) and other applications.

This system is used in police training sessions in coordination with the Starkville, MS Police Department's Special Weapons and Tactics team (SWAT). Once a month, the SWAT team spends time at MSU training in the Robotics Testbed at the Center for Advanced Vehicular Systems (CAVS). The Robotics Testbed was designed to be a re-configurable space and, for the purpose of law enforcement operations training, currently houses a mock 2 bedroom apartment. The mock apartment is used by the SWAT team to train officers to properly investigate and clear a building while facing hostile assailants. The Robotics Testbed supports the use of simulated ammunition that allow officers to operate real weapons in a non-lethal manner (using chalk rounds). This capability allows for very realistic training scenarios.

Training with the robot platform and the integrated voice system revealed that, overall, officers had difficulty remembering the specific keywords for the many commands available for use with the system [10]. Their inability to recall the commands led to verbal expressions of frustration by the officers. Although they were frustrated by the current implementation, they saw potential for the voice integration. It was therefore a priority to design a tool that could be used by the officers to train on the words available to them when using the robot. However, the training tool needed to be independent of the robotic platform so that the officers did not have to be co-located with the system in order to train. This requirement for independence from the robot led to the development of the web-based training tool discussed in [11].

The web-based training tool was created by integrating the Julius voice recognition system into training protocols that were hosted on the web. Julius interpreted commands spoken into a microphone. The commands were written to file and a web server processed the output provided by Julius. Voice recordings were saved for each person training with the tool. The concept was to provide the officers with a training tool, which employed a prompt and response system. Officers were prompted to speak a command while the system was listening. When the officers spoke the displayed command, the processed results were displayed on the screen. The system allowed three attempts at saying the specified command. After each attempt the results were logged by the system. After three attempts, the user was allowed to skip the prompt and continue to the next prompt. At the end of a training session, the officer could review a file summarizing which words were skipped, how many attempts each word took, and which phrases were misinterpreted.

The web-based tool was presented as an option to train on the voice integration of Julius with the robotic systems, such that users could train on how to properly interact with the system. This allowed them to learn how to pronounce the commands (the system had some difficulty with the many different accents present in the southern United States). However, through internal testing, it became clear that while the web-based tool provided a means for users to learn to speak in a manner such that the system could understand them, the initial goal of providing a training tool that could help officers recall the specific commands was not necessarily being met. This was due to the fact that beyond text-based confirmation, the web-based tool provided nothing that allowed the officers to associate the spoken commands with specific actions to be performed by the robot. In order to resolve this issue, the tool was redesigned to work with a virtual environment, so that the users could see the results of their spoken commands on a simulated robotic platform. The goal was to allow officers to train on a robot that executed the spoken commands but still did not require the officers to have a physical robotic system.

IV. VR Implementation

The virtual training tool was implemented using the Unity game engine. Unity is cross-platform and provides support for 3D virtual environments with easy integration of head mounted displays [12]. The Julius voice recognition software was integrated with Unity using network sockets. Julius automatically captured the default microphone of the system at start-up. Parsed commands were passed to a loop that listens exclusively for commands to be passed by Julius to the system for processing. A look-up table links command strings with a handler function that implemented the action. This implementation allowed for easy expansion of capabilities in the system by simply adding new strings and new functions to the look-up table.

Initially, users were presented with a menu that allowed them to select from a variety of different options for customizing the training scenario. First, the user may select an environment for the training scenario. In the initial implementation of the system, two environments were provided: a bank and a school. The bank environment consisted of a single room with multiple cubicles. In the bank, a person was hidden between two sets of cubicles. The user could drive the robot through the room and verbally issue commands to the robot as they searched for the person. The bank environment can be seen in Figure 2.

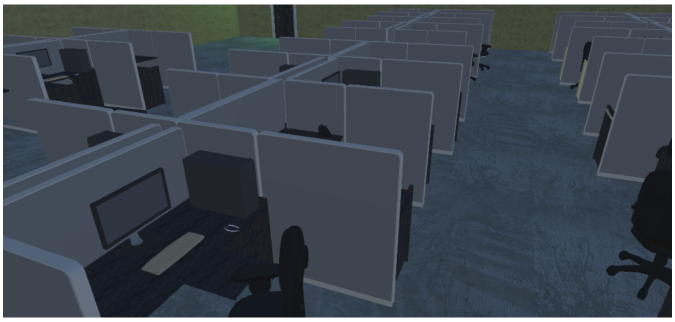

Fig. 2. Bank Scene Virtual Environment

The second environment provided was a school hallway. The school environment consisted of a long hallway with lockers and two classrooms. The first room was located on the right side of the hallway and did not contain a person. The rooms were populated with typical classroom materials such as blackboards, desks, and chairs. The second room at the end of the hall on the left was an exact replica of the first room except that it included a person standing in the center of the room with their hands raised in the air. The school environment can be seen in Figure 3.

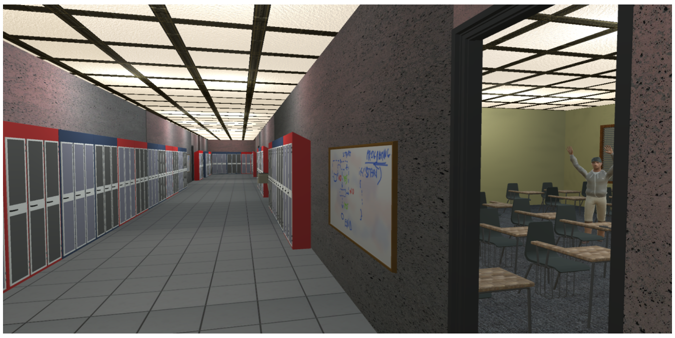

Fig. 3. School Scene Virtual Environment-second room from the perspective to see person standing in the room and hallway the robot traverses.

Once the user selected an environment, the user was prompted to choose a robotic system to use in the training scenario. The initial implementation provided the user with a single choice, a model of the Throwbot XT by Recon Robotics [13]; however future iterations will have other robots available. This robot is a small tactical platform. Within the virtual environment, the robot's capabilities were expanded to include systems not typically available on the real-world version of the robotic system. In order to test specific commands, a spotlight and strobe lights were added to the robot system. Additionally, an audio speaker was attached. None of this specific hardware was represented in the robot's geometric model and was only associated with specific locations on the robot for the generation of realistic lights and sounds.

Once a robot was selected, the user specified which equipment they wanted to use for a visual display. In the training scenarios, the visual display provided a secondary interface to display the list of available commands, what Julius heard when the user was talking, and prompts to confirm commands. In the initial implementation, the user could select a virtual implementation of a Google Glass optical head-mounted display. Google Glass is an augmented reality head mounted display system that effectively provides the user with a heads-up display. On this display (see Figure 5), the available commands are listed as well as their outcomes. The available commands can be seen in Table I.

The user was then provided options for how to interact with Julius. Three options were initially provided. The first option was to simply *display the command* that was interpreted by the system and then execute that command. This was the mode of operation that was used by the physical system. This mode of operation can lead to problems when Julius misinterprets the user's speech or picks up background noise as input. We felt it was important to include this mode of interaction as it was what officers initially interacted with during training [10]. The disadvantage of this approach was that analysis of the interaction required manual review of the entire interaction and any faulty interpretations were then marked for future improvement of the system.

Current Commands		
Keyword	**Command**	**Result**
Apple	Light	Turns off all lights
Apple	Dog	Plays dog barking
Apple	Spot	Turns on spot light
Apple	Dark	Turns off spot light
Apple	Strobe	Turns on strobe light
Apple	Off	Turns off strobe light
Apple	Flash	Turns on blue/red lights
Apple	Siren	Plays siren sound
Apple	Quiet	Stops all sounds

TABLE I
TABLE OF COMMANDS AVAILABLE TO USE DURING TRAINING

The second interaction mode was the *press to confirm* option. When this option was selected, the Julius system interpreted a spoken command, and then the interpreted text was presented to the user on the display, as seen in Figure 5. The user was prompted to confirm the interpreted command by pressing Enter on the keyboard. When the user confirmed the command, it would be executed. This mode of interaction was one that was suggested based on verbal feedback from officers that interacted with the voice recognition system when used with a physical robot. This allowed the user to filter out any misinterpretations that occurred and ensure that lights or sounds would not be accidentally activated by the system when it was not desired. An advantage of this option was that the system could automatically evaluate the number of confirmed and not confirmed commands after a scenario was completed. Based on the captured recordings and timestamps, the system could be adapted for specific users and/or the faulty interactions could be analyzed.

Finally, the third interaction method provided to the users was *press to talk*. In this mode, the user would have to press and hold the Enter key to notify the system that it wanted to receive input from Julius (see Figure 4). Any interpretations provided by Julius when the Enter key was not pressed where simply thrown out. However, any commands interpreted while the key was pressed would be executed immediately. This mode of interaction provided the ability to filter out any interpretations that occurred when the user didn't want the system to execute a command. It also provided fast execution of commands when the user wanted to issue commands, but, provided very little control over what was executed once the system was listening. When using this method, a wrong command could be interpreted and executed while the press to talk was activated. An advantage of this approach was that all recordings with a timestamp outside the pressed button area could be deleted after interaction automatically.

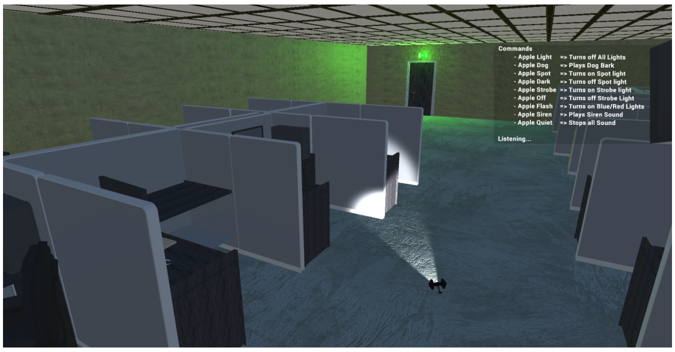

Fig. 4. Press to talk prompt displayed in the VR Oculus Headset

Two versions of the software were implemented. One version was designed to be used exclusively on a desktop when the user did not have access to a Virtual Reality headset. The user operated the robotic system by using the WASD keys. The simulated Google Glass display was presented in an approximately correct location for the provided viewpoint. A chase camera was attached to the robotic system and moved with the robot through the environment. Voice input was provided to the system using a microphone attached to the computer. A screenshot of this interaction method can be seen in Figure 5.

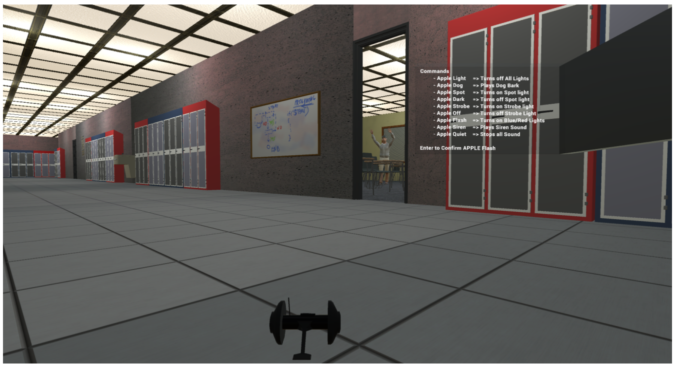

Fig. 5. Press to confirm prompt in desktop view

The second version was designed for use in Virtual Reality using an Oculus Rift head-mounted display [14]. An Oculus Rift is an immersive Virtual Reality head mounted display, which integrates easily with Unity. In this version of the software,the user's perspective was independent of the robotic system. This allowed for increased sense of realism and presence in the environment allowing the user to feel as if they were standing within the scene. The robotic system was driven using the Oculus Touch controllers [14]. The movement of the robot was tied to the left thumbstick. In the Virtual Reality version, the Google Glass display was positioned in front of the person's right eye, so that the user felt as if they were wearing the glasses. The VR version used the Oculus Rift's built-in microphone to capture the commands spoken by the user. Confirmation of a spoken command was provided by pressing the "A" button on the Oculus Rift Touch controllers.

V. Conclusion

We expect the revised training tool to improve officer recall of verbal commands through association of the commands with actions observed in the virtual environment when interacting with the robotic system using the Julius voice recognition system. In addition to using the platform for training interaction with the voice system, the training tool can be easily expanded to allow for prototyping and testing different interfaces for the Julius system and for the general robotic system. By leveraging the Unity development platform, this tool provides the ability to be easily expandable. We have shown that multiple environment scenarios can be designed and implemented with ease. A multitude of real-world interfaces can be implemented within the virtual environment for use, testing, and evaluation. Changing out robotic platforms or vehicles for testing the use of different interfaces and voice recognition is relatively simple. Additionally, the tool provides the option for a high level of immersion for the participant by leveraging Unity's integration with VR head-mounted displays such as the Oculus Rift.

VI. Future Directions

The virtual training tool has the ability to be expanded in many different ways. Additional environments, including dynamic elements, need to be implemented to allow for a greater variety in training scenarios. Furthermore, additional interfaces for displaying the input provided to Julius should be tested within the environment to help users decide which interfaces make sense for their use of Julius. The effectiveness of the tool should be formally evaluated through comparison of user performance with the robot using the web-based training tool and the VR-based training tool. Additionally, performance of users with virtual interfaces should be compared to performance with real interfaces. Finally, expanding the virtual training tool to include a larger set of languages will support the wider adoption of the tool and should provide for a larger user base internationally. TUKE is planning to demonstrate the tool on International Museum Day (May 18) in Kosice and evaluate the interactions in both English and Slovak languages.

Acknowledgment

The research presented in this paper was partially supported by the Center for Advanced Vehicular Systems and partially by the Slovak Research and Development Agency under the contract No. APVV-15-0731 & APVV SK-TW-2017-0005.

References

[1] A. Lee, T. Kawahara, K. Shikano, *Julius an open source realtime large vocabulary recognition engine*, In proceedings: EUROSPEECH, 7th European Conference on Speech Communication and Technology, September 3 - 7, 2001, Aalborg Congress and Culture Centre, Aalborg, Denmark, pp. 1691–1694, 2001.
[2] A. Mossel, A. Peer, M. Froeschl, J. Goellner, C. Schoenauer, H. Kaufmann, *VROnSite: Towards Immersive Training of First Responder Squad Leaders in Untethered Virtual Reality*, IEEE Virtual Reality, 2017.
[3] A. Mossel, A. Peer, J. Goellner, H. Kaufmann, *Requirements Analysis on a Virtual Reality Training System for CBRN Crisis Preparedness*, Proceedings of the 59th Annual Meeting of the International Society for the Systems Sciences, 1 (1), 2015.

[4] F. Buttussi, L. Chittaro, *Effects of Different Types of Virtual Reality Display on Presence and Learning in a Safety Training Scenario*, IEEE Transactions on Visualization and Computer Graphics, 2017.

[5] P. Khooshabeh, I. Choromanski, C. Neubauer, D. M. Krum, R. Spicer, J. Campbell, *Mixed Reality Training for Tank Platoon Leader Communication Skills*, IEEE Virtual Reality, 2017.

[6] R. Feloni, *Walmart is using virtual reality to train its employees*, 2017. [Online]. Available: http://www.businessinsider.com/walmart-usingvirtual-reality-employee-training-2017-6. [Accessed: 6-April-2018].

[7] K. Vertanen, *Baseline WSJ acoustic models for HTK and Sphinx: Training recipes and recognition experiments*, Technical report, Cambridge, United Kingdom: Cavendish Laboratory, 2006.

[8] V. Zue, S. Seneff, and J. Glass, *Speech database development at MIT: TIMIT and beyond.* Speech Communication, 9(4), pp. 351–356, 1990.

[9] M. Pleva, J. Juhar, *TUKE-BNews-SK: Slovak Broadcast News Corpus Construction and Evaluation*, In proceedings: LREC 2014, Ninth International Conference on Language Resources and Evaluation, ELRA, Reykjavik, pp. 1709–1713, 2014.

[10] M. Pleva, J. Juhar, A. Cizmar, C. Hudson, D. Carruth, C. Bethel, *Implementing English speech interface to Jaguar robot for SWAT training*, In proceedings: Applied Machine Intelligence and Informatics (SAMI), 2017 IEEE 15th International Symposium on, Herlany, Slovakia, pp. 105–110, 2017.

[11] C. Hudson, D. Carruth, C. Bethel, M. Pleva, J. Juhar, S. Ondas *A Training Tool for Speech Driven Human-Robot Interaction Applications*, In Emerging eLearning Technologies and Applications (ICETA), 2017 15th International Conference on, Slovakia, pp. 1-6. IEEE, 2017.

[12] *Unity Game Engine*. Available from http://www.unity3d.com. [Accessed: 6-April-2018].

[13] Recon Robotics. *Recon Robotics-Throwbot XT with Audio.* Available from http://www.reconrobotics.com/products/throwbot-xt/ [Accessed: 6-April-2018].

[14] Oculus, V.R. *Oculus rift*. Available from https://www.oculus.com/ [Accessed: 6-April-2018].