Vocal One Switch (VOS) selection interfaces for Virtual and Augmented Reality hands-free tasks

STAG: Smart Tools and Apps in computer Graphics (2016)
Giovanni Pintore and Filippo Stanco (Editors)

Vocal One Switch (VOS) Selection Interfaces for Virtual and Augmented Reality Hands-free Tasks

F. Manuri, A. Sanna, F. Lamberti and G. Paravati

Dipartimento di Automatica e Informatica, Politecnico di Torino, Torino, Italy

Abstract

Several virtual and augmented reality tasks involve users in hands-free interactions; in this case, speech-based systems are often preferred for their intuitiveness and naturalness. On the other hand, the robustness of this kind of interfaces can be an issue, thus affecting both the usability and the user experience, when they are used in noisy environments. This paper proposes a comparison of a traditional multiword interface with a one switch interface triggered by vocal commands: three different scanning algorithms are tested. With one switch scanning interfaces users can sequentially select the desired command, thus improving the robustness of traditional multiword speech recognition-based interfaces. Latency time is an issue for one switch interfaces, but it is shown how a bidirectional scanning algorithm based on a non binary switch can strongly mitigate this problem. The comparison considered both objectively (robustness and efficiency) and subjectively (user feedback) parameters.

Categories and Subject Descriptors (according to ACM CCS): H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities. H.5.2 [Information Interfaces and Presentation]: User Interfaces—Interaction styles.

1. Introduction

Virtual and Augmented reality applications usually provide very sophisticated and efficient interfaces based on different input modes, which are often used together to deploy multimodal user interfaces (MUIs). On the other hand, hands-free tasks (e.g., maintenance and assembly) can benefit neither of solutions based on touch, gestures and poses nor take advantage of haptic-tangible interfaces; moreover, wearable systems might further limit possible input modes.

Gaze and speech-based interactions are widespread to tackle potential issues related to hands-free tasks. Although gazing is a very expressive and natural form of human interaction/communication, a special purpose hardware is usually necessary to implement robust interfaces; moreover, gaze tracking devices can prevent the use of other devices such us VR & AR glasses. Speech-recognition systems are now able to provide extremely high ratios in correctly recognizing vocal commands and verbal communication is one of the most powerful and expressive forms. Unfortunately, performance of speech recognition systems are strongly affected by noise. The environmental noise of an industrial plant as well as the hubbub in extremely populated environments might make speech recognition-based UIs unusable.

Robustness and usability issues have been deeply investigated also in different contexts. In particular, the design of interfaces for impaired people (people with severe motor or cerebral disabilities) was focused on alternative paradigms to input commands that have to be based on extremely simple, robust and intuitive solutions. A lot of examples of interfaces for impaired people are based on scanning algorithms that present available commands in a sequential way; the user cannot choose the desired command in a random way and a certain latency time, in general, occurs. On the other hand, a command selection is possible by means of a simple binary switch, thus considerably reducing the complexity of the interface. This kind of interfaces is usually named one switch or single switch.

This work proposes and assesses a hybrid solution for accomplishing VR & AR tasks to be performed hands-free. A traditional speech recognition-based interface has been modified in order to present selectable commands by scanning algorithms. As vocal commands can be configured as a non-binary switch, an efficient solution based on a three-words interface (equivalent to a three-state switch) allows users to efficiently and robustly select the desired command. With respect to a traditional speech interface based on a dictionary (which allows a random access to commands), the proposed solution is slower, but it overcomes traditional robustness problems of speech interfaces when used in noisy environments.

Three different scanning algorithms implemented for a VOS interface have been tested: automatic, inverse and bidirectional. Objective results show how one switch solutions can overcome robustness problems related to traditional speech recognition-based interfaces when used in noisy environments. Moreover, the bidirectional scanning algorithm reduces considerably latency scanning

time and four users of eleven participating to tests preferred it with respect to all the other solutions. Although these positive feedback, the bidirectional scanning algorithm is still affected by the number of words to be pronounced.

The paper is organized as follows: Section 2 reviews both input modalities for VR & AR applications and one-switch interfaces, Section 3 presents the proposed interface and Section 4 describes how the tests have been performed and the obtained results.

2. Previous work

This Section describes both possible input modalities in virtual-augmented reality applications and reviews one switch interfaces as well as some existing implementations.

2.1. Input interaction modalities

Virtual and augmented reality applications use several input modalities. Traditional input devices such as keyboard, mouse and joystick have been soon replaced by more sophisticated and natural interaction ways. For instance, touch and multi-touch surfaces are used to navigate virtual worlds [KGMQ09] as well as to remotely manipulate parts of a robot by an augmented interface [HIII13]. Hands and body gestures can be an efficient and intuitive way to convey inputs; wearable [MP15] [MM09] [MMC09] as well as "desktop" solutions [RCH13] [SLBM16] [DBR13] based on commercial tracking devices such as the Leap Motion or the Microsoft Kinect are able to use gestures and poses to navigate and, more in general, to interact with virtual and augmented environments. Tangible interfaces are another well known input solution for VR & AR applications; tangible interfaces are well suited for manipulation tasks [LNBK04] [BKP01] [BKP08]. When any form of hands-based interaction cannot be used, an alternative interaction technique is necessary; in this case, gaze and speech are usually considered. For instance, gaze interaction is considered in [PLC08] and [OYT96], whereas vocal commands are used in [LMP*16], [BM06] and [GSZN03]. Also, brain interfaces have been profitably applied to VR & AR worlds (see for instance [LLR*08] and [Nav04]). In order to tackle issues such as robustness and usability, two or more of the above mentioned input interaction modes are used concurrently, thus implementing the so called multimodal interfaces [DBH*09], [HBK07], [OBF03], [BRC96], [GWZ97] and [KVL07].

2.2. One switch interfaces

One switch (or single switch) interfaces have been deeply investigated in order to support the interface design for people with different kinds of disability. A traditional interface provides a direct selection paradigm, thus enabling users to activate any available command; in other words, a sort of "random access command" modality is supported. This kind of access requires a great level of interaction between user and machine, which is not available for people with severe motor or cerebral disabilities. One switch interfaces try to overcome this issue by presenting available commands in a sequential way: the user can activate a desired command by pressing a button, by a vocal input, by blinking or by any sort of input that can be assimilated to a switch activation.

It is immediately clear how the scanning algorithm, which presents sequentially available commands, is a key issue for the interface usability; in particular, the scanning latency (also called scanning delay) has to be accurately tuned. Different scanning algorithms can be implemented and they are categorized as [Ang92]:

1. regular or automatic - selectable elements are scanned cyclically and the user selects the desired command when highlighted;
2. inverse - the scanning selection advances only when the "switch" is continuously activated and the user can select a command by releasing the switch on it;
3. step - successive (discrete) switch triggers allow the user to select the desired command.

One switch selection interfaces are used in different applications, usually to improve the daily life of impaired people, ranging from text entry [BCM09] to video games [ALCDR15]. For instance, an adaptive scanning algorithm is proposed in [SK99] to efficiently perform text entry tasks, whereas a robotic arm is controlled by a single switch user interface to support people with less muscular strength in [WYNC09]. Wheelchairs can be driven by single switch interfaces [YG98] and a single switch scanning interface is used in [GB10] to allow people with amyotrophic lateral sclerosis to control a keyboard by eye control. Mouse manipulation has been implemented by a single switch solution in [BASQ*04], whereas more complex human-computer interactions are proposed in [BR08] and [TT87]; in [BR08] objects on the screen are clustered by a sort of quad-tree algorithm in order to speed-up their selection, whereas a scanning keyboard has been implemented in [TT87] to allow children who are severely physically disabled to access microcomputers for writing, playing, and engaging in educational activities. Internet navigation is also available: a web browser controlled by a single switch interface has been presented in [Raj04]. Although all these solutions based on scanning algorithms are considered slower than direct-selection applications, they usually provide a more robust interaction.

3. Proposed solution

The idea of this work is to use a speech-based interface to trigger a scanning selection; both virtual and augmented reality applications, where other input modalities are prevented, can take advantage of the proposed solution. For instance, wearable AR-based applications for maintenance allow technicians to perform hands-free tasks [Wil96] and neither touch-based nor touch-less input paradigms can be used. On the other hand, speech recognition, considered natural and intuitive, might be not robust enough to be used in noisy environments. For this reason a sort of hybrid input speech interface is proposed: commands are not directly selectable but are sequentially activable by a scanning algorithm. An automatic scanning and an inverse scanning algorithm controlled by a single word have been implemented. Moreover, as a vocal switch is not necessarily binary as a physical one, a bidirectional scanning algorithm is also presented. In this case, two words are used to scan the command list and a third word is used to select the desired command. The goal of this work is to compare possible one switch scanning interfaces with a traditional multiword speech-based solution and obtain some indications about usability in noisy environments.

The target application developed for the considered use case

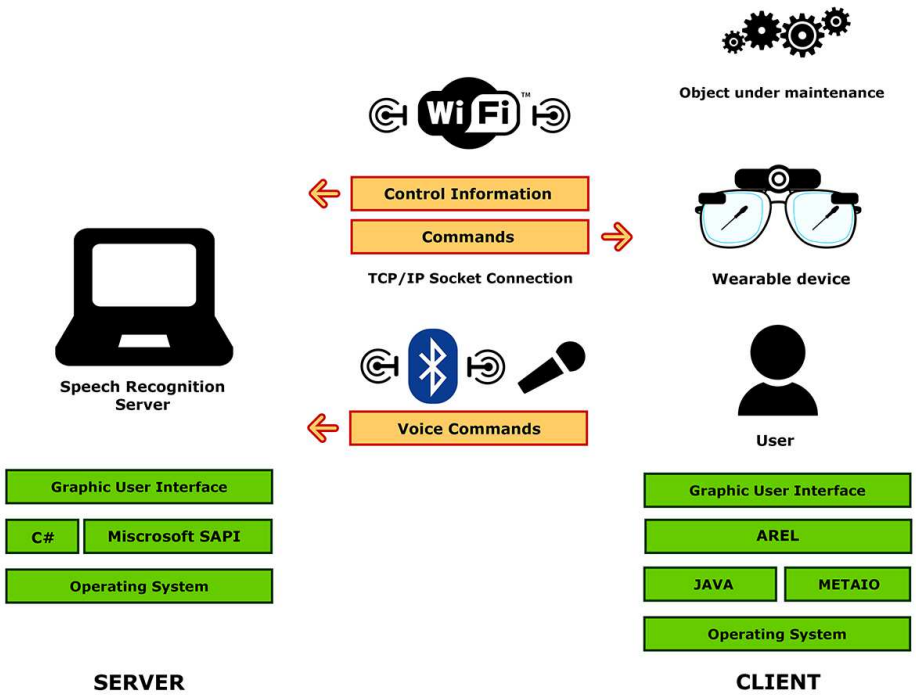

Figure 1: *Architecture of the proposed framework.*

[SMP*15] is an AR application for performing maintenance procedures. The use case requires the user to complete a sequence of steps in order to accomplish a maintenance task and the application provides assistance to him/her through AR contents. Since each step requires the user to perform operations that potentially involve the use of both hands, such as unscrewing bolts or removing components from a machinery, a hands-free interface is necessary. At each step of the procedure, a set of icons is used to inform the user on the available functionality provided by the application, as showed in Fig. 2.

Wearable devices present some limitations to the deploying of an application that involves tracking algorithms, graphic resources for displaying AR contents and a speech recognition system to provide the user interaction. The computing power may be inadequate to process smoothly all the required resources and the high computing tasks may dramatically reduce the battery life of the device, thus making it impossible to complete the given task. Moreover, the libraries for speech recognition available for wearable devices are not flexible and responsive as the ones available for desktop environment, especially in terms of robustness and languages supported.

For these reasons, the speech recognition system has been developed on a desktop system and it works as the server side of the proposed solution, returning the recognized commands to the AR application running on the wearable device.

The used framework consists of a client-server architecture, as showed in Fig. 1. The user pronounces commands into a Bluetooth microphone directly connected to the server. The server handles the speech recognition and communicates to the client the functionality to be activated based on the uttered command. Then the client activates the complementary functionality and, if a change to the client interface occurs, it notifies it to the server in order to load the corresponding set of commands. The communication between client and server occurs on a local Wi-Fi network, through a socket connection.

The client side of the system consists of an AR application for AR-Glasses that manages the user interface and the communications with the server. The AREL technology has been adopted to build the application, using the Metaio SDK (`https://en.wikipedia.org/wiki/Metaio`) AR framework: this solution allows developers to create the user interface as an HTML page with the logic defined in JavaScript, in the form of a library.

The server side consists of a C# application that takes care of the speech recognition of the commands for each state of the user interface. First of all, the server loads a state machine representation of the client user interface, provided in an extension of the SCXML notation as presented in [LMP*16]. The state machine is expected to describe the layout of the client interface, as each state lists the available functionalities and how they modify the interface current state when activated. Moreover, the specific set of words to be recognized and mapped to a specific functionality is defined. Whenever a functionality compels the UI to change to another state, the server is notified and the corresponding set of words to be recognized is loaded. The speech recognition module has been developed using the Microsoft Speech Platform (`https://msdn.microsoft.com/en-us/library/jj127860.aspx`), which sup-

Figure 2: *A screenshot of the application user interface.*

ports 26 different languages. The current implementation of the system operates in the Italian language, for the sake of reducing recognition errors caused by mispronunciation.

At the beginning, the application asks users for choosing among the four different available interfaces: multi-word speech recognition-based system (MW), a one switch interface based on an automatic scanning algorithm (AVOS), a one switch interface based on an inverse scanning algorithm (IVOS) and a one switch interface based on a bidirectional scanning algorithm (BVOS). Then, the JavaScript library connects to the server and creates the user interface for the first state, displaying the corresponding icons, and finally it notifies the server. When a state update for the UI is received, the server loads the corresponding vocabulary and grammar. Every time a command is recognized with enough confidence, the client is notified that the matching functionality is invoked, thus activating it. Moreover, as the server provides information on the level of confidence when recognizing a command, a colored rectangle is shown in the top right corner of the UI, in order to provide a visual feedback to the user actions. The rectangle will assume three different colors, depending on the degree of confidence in the recognition phase:

1. green if the command was correctly recognized;
2. yellow if the command was present in the current set of recognizable words but the level of confidence was too low;
3. red if the pronounced command was completely unintelligible.

Moreover, each time an icon is activated, the background is set to transparent and a blue border appears for 500 ms to give a visual feedback to the user. The icons corresponding to the available functionalities are displaced on the left and right side of the interface, in a circular buffer: when the last icon is reached the following one will be the first one, and vice versa.

The four different interfaces provide the following interaction systems:

1. MW. In this modality, when a command is correctly recognized, the corresponding icon is highlighted with a blue border.
2. AVOS. In this case, the icons are highlighted one at a time with a green background and a latency of 4500 ms and if the user pronounces the confirmation command, the currently highlighted icon is activated. After 3000 ms, the icon background shades to red to advise the user that it is too late to start pronouncing the command, as the following icon would be activated. Finally, the background is turned off for the remaining 500 ms, before activating the following icon. The total latency was determined after numerous tests to provide the best trade-off between the waiting time for the user (to be minimized) and the time necessary to pronounce the word, process it on the server and provide a feedback to the client if correctly recognized (in a reasonable time).
3. IVOS. In this case, when a new state of the UI is loaded, the first icon is highlighted with a green background. When the user pronounces the command to advance, the background of the current icon is turned off, and the next one is highlighted. If the user highlights an icon and then he/she does not pronounce the command for 4500 ms, the action corresponding to the current icon is activated. After 3000 ms, the icon background shades to red to advise the user that it is too late to start pronouncing the command, as the current icon is going to be activated. Finally, the background is turned off and a blue border appears to advise the user that the current icon is being activated. The total latency time was determined in the same way as for AVOS.
4. BVOS. In this case, when a new state of the UI is loaded, the first icon is highlighted with a green background. The user can then activate the current icon with the confirmation command, or move to the previous or next icon with the specific command, thus moving the highlighting to another icon.

4. Tests

This Section presents and discusses the obtained results both from an objective and subjective point of view. Eleven people tested the four interfaces: MW, AVOS, IVOS and BVOS. The first two scanning algorithms are triggered by a single word (the equivalent of a physical switch), whereas the last scanning algorithm is based on a three-words solution, where two words are used to move forward and backward into a list of commands organized as a circular buffer and the third word is used to confirm a command selection. These are the words used in three scanning algorithms: for AVOS the Italian command is “conferma", equivalent to the English word ‘confirm’; for IVOS the Italian command is “avanti", equivalent to the English word ‘next’; for BVOS, the Italian commands are “avanti", “indietro" and “conferma", respectively equivalent to the English words ‘next’, ‘previous’ and ‘confirm’.

4.1. User Test

The test requires the users to interact with an AR application for maintenance operations. The users have to navigate throughout the menus of the application and activate icons in order to try all the available functionalities. A sequence of slides displayed on a monitor instructs the users on which icon they should activate, step by step. Each user had to repeat the test four times in order to try out all the four different solutions. The users that participated to the test were both males (9) and females (2) students of the MSc in computer science at the Politecnico di Torino. Their age ranged between twentyfive and thirtythree. The users mostly declared to possess an average competence in the use of speak recognition interfaces.

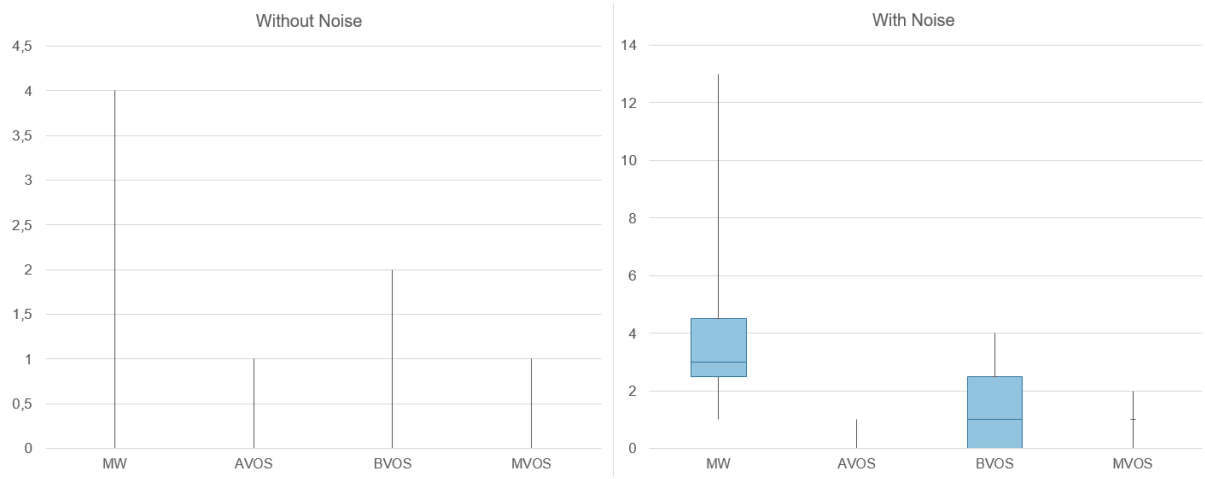

Figure 3: *Number of errors (false negative + false positive) with and without environmental noise.*

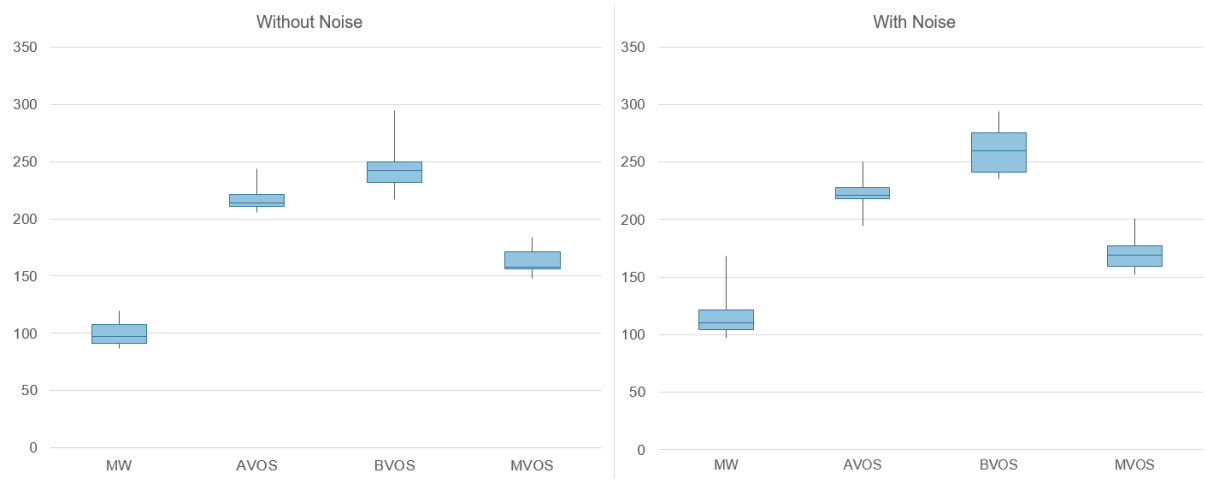

Figure 4: *Times (in seconds) to complete the assigned task.*

4.2. Methodology

The eleven testers were trained individually; in particular, they were asked for performing a hands-free maintenance task. The number of steps has been previously defined and was kept equal for all tests. As testers were not professional technicians but students of the MSc in computer science, a table of vocal commands related to each icon of the interface was provided; in this way, it has been avoided to artificially increase the mental load when tests with the MW interface were performed.

Each tester tried all four solutions and then filled a questionnaire (more details in Section 4.4). The same interface was first tested in a quiet environment and then the test was repeated by adding an artificial environmental noise. The artificial noise was aimed to simulate the background noise in an industrial plant; the average intensity of the noise was approximately 67dB, with a maximum recorded value of 74dB. For each test, the noise track was played from the beginning in order to provide the same conditions.

Some objective parameters were recorded: the number of false positive, the number of false negative, the number of words pronounced and the time necessary to complete the task. A false positive is considered when a wrong command is triggered; this can happen when the recognition engine confuses an environmental "sound" as a valid command or when the scanning algorithm leads to select a wrong command (this is possible, for instance, for scanning algorithms based on temporized selection mechanisms). When a right command is not recognized (for instance, when a loud background noise temporarily overlaps) the number of false negative increases.

4.3. Objective results

Number of errors, completion times and number of pronounced words are listed in Fig. 3, 4 and 5, respectively. It is possible to notice how the performance of all the interfaces drops with the environmental noise; on the other hand, the three VOS interfaces

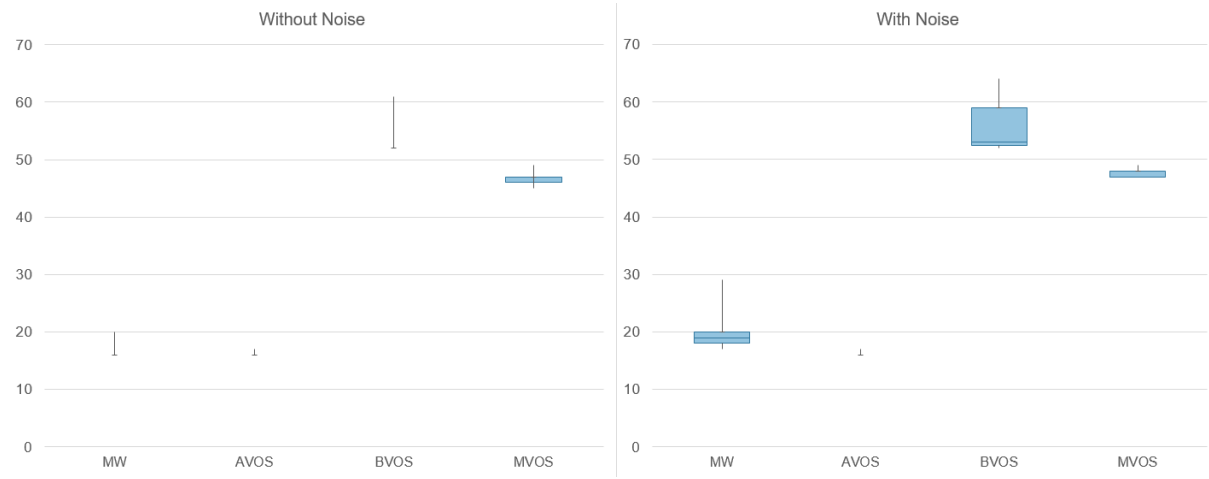

Figure 5: *Words pronounced to complete the assigned task.*

limit the number of errors to a maximum of three. Also, a statistical analysis confirm this claim. As variances are unknown, paired t-tests are used to test the null hypothesis that the mean difference between number of errors of the MW interface and each VOS interface is equal to zero (e.g., $\mu_t = \mu_{MW} - \mu_{AVOS/IVOS/BVOS} = 0$). On the other hand, the alternative hypothesis is that the robustness of VOS interfaces is better. A level of significance has been considered: $\alpha = 0.05$. T-statistic values show how the null hypothesis can be always rejected: when MW and AVOS are compared, statistic t is equal to 3,89 with respect to a t critical equals to 2,22, whereas statistic t is equal to 2,26 with respect to a t critical of 2,13 when MW and IVOS are compared and statistic t equals to 2,90 with respect to a critical value of 2,20 when MW is compared with BVOS. The same analysis about completion times and number of words pronounced outlines how the MW interface is always better of the scanning algorithm-based solutions considered; the only exception is the AVOS interface when the number of pronounced words is considered.

If the robustness is improved by VOS interfaces, the latency time can be an issue. Automatic and inverse scanning algorithms lead to an average completion time about the double of the MW solution. On the other hand, the bidirectional scanning algorithm provides an increased robustness and a limited overhead in latency times. As better outlined in Section 4.4, the BVOS interface is affected by a number of words to be pronounced that is about the triple of the MW interface. From an objective point of view, it is not easy to definitely select the best approach as a lot of other parameters should be also considered. Robustness and latency are just two dimensions of a domain where mental load, user preferences, environmental conditions and the application itself play a non marginal role.

As mentioned before, the list of commands has been provided, but this might strongly reduce the mental load really necessary to use the application. Moreover, the proposed application can be controlled by a very limited number of commands (less than 20); very different performance could be detected for more complex applications presenting several tens commands: the robustness of speech recognition-based applications generally decreases with the number of words. VOS interfaces are not affected in term of robustness as the dictionary size is constant. On the other hand, a larger number of commands entails a larger scanning (latency) time and this issue should be tackled by considering more sophisticated scanning algorithms such as the ones introduced for text entry [BCM09].

4.4. Subjective results

After completing the test, the testers were asked to evaluate their experience in three different ways. Firstly, they had to rate the interfaces in a scale between 0 (bad) and 4 (good) for five different qualities to evaluate the usability of the interfaces, as defined in [Nie96]. The five qualities were defined as follows:

1. Learnability. "How easy is it for you to follow the proposed instructions the first time you encounter the interface?"
2. Efficiency/Effectiveness. "Once you have learned what to do, how quickly can you perform the proposed instructions?"
3. Memorability. "When returning to the application after a period without using it, how easily would you reestablish proficiency?"
4. Errors. "How many errors did you make, how severe were these errors and how easily did you recover from them?"
5. Satisfaction. "How pleasant is it to use the interface?"

The overall results of the Nielsen usability questionnaire are listed in Fig. 6. The BVOS interface has the higher rating for learnability, memorability and errors, and it is only second to the MW interface for efficiency/effectiveness and satisfaction. Overall, it is the interface with the higher evaluation. The MW interface got the best evaluation for both efficiency and satisfaction, but it got the worst results for learnability and memorability, achieving the second place among the four available interfaces. The AVOS and IVOS are considered better than the MW for learnability and memorability but they are otherwise considered the worst solutions, with a slight overall difference between them.

Secondly, the testers had to rate the interfaces, on a scale be-

Table 1: *Ranking of the four interfaces, expressed as number of testers that choose each option.*

Interface	First Choice	Second Choice	Third Choice	Fourth Choice
MW	7	4	0	0
AVOS	0	1	2	8
IVOS	0	0	8	3
BVOS	4	6	1	0

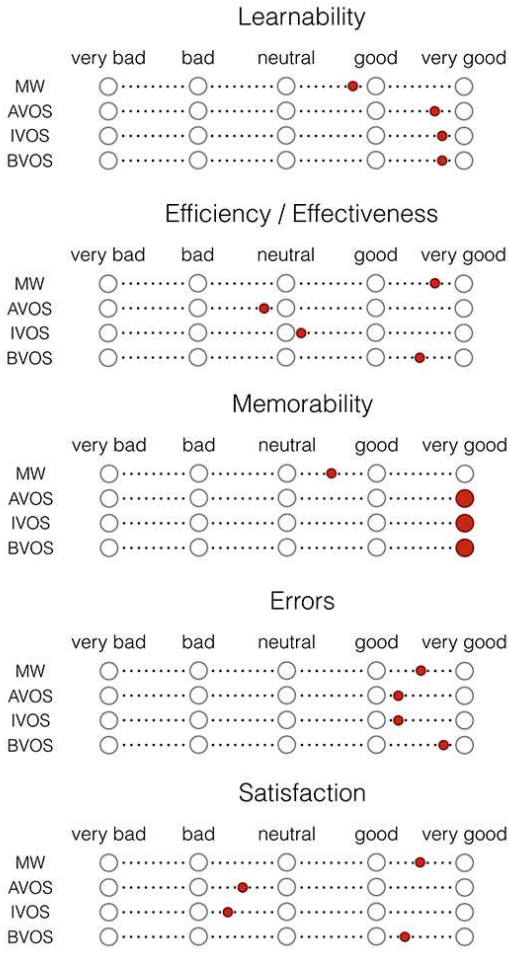

Figure 6: *Usability evaluation based on the NIELSEN usability principles.*

tween 0 (bad) and 4 (good), for six different qualities derived from the Subjective Assessment of Speech-System Interface Usability (SASSI) principles: "system response accuracy", "likeability", "cognitive demand", "annoyance", "habitability" and "speed" (terms as defined in [HG00]). These six qualities were described to the users as follows:

1. "system response accuracy" refers to the robustness of the system in recognizing the user's input correctly and whether the system does what the user expects;
2. "likeability" means that the users enjoy using the system, perceive the system as friendly and would use it again;
3. "cognitive demand" refers to how much difficult and stressful the system is to be used and how much effort and concentration it requires;
4. "annoyance" is related to how much the system is irritating/repetitive/boring;
5. "habitability" defines the user's confidence in what the system is doing and how to interact with it;
6. "speed" simply refers to the speed of the system.

The overall results of the second usability questionnaire are listed in Fig. 7. The MW interface is perceived as the overall best solution, even if by a slight margin, and it is the better one in terms of likeability, speed and minimum annoyance. It is considered worse than the BVOS in terms of accuracy and habitability, thus the BVOS is perceived as the second best option among the four available interfaces. The AVOS and IVOS interfaces are considered better than the other two only in terms of cognitive demands, thus they are classified by the users as the worst possible solutions, with a minor difference in the overall evaluation between the two.

Finally, the users where asked to rank the four proposed interfaces and to provide a motivation for their choices. The results are showed in Table 1. Seven testers out of eleven selected the MW interface as their first choice, because it is the fastest and easiest interface available, with the lowest latency value and the lowest number of commands to pronounce. Four testers selected the BVOS interface as their first choice despite of the high number of commands they have to utter, because they perceived the need to look for the correspondence between the icons and the vocal command or to learn it as a limitation. Three users depicted the BVOS system as the best alternate solution to the MW interface in terms of better reliability and lesser cognitive demand. Only one tester selected the AVOS as a valid alternative to the MW interface due to its simplicity. The AVOS and IVOS solutions were generally depicted as the worst interfaces in view of their high latency time. Two users preferred the AVOS considering its easiness. Eight users preferred the IVOS because they had more control on the interaction with the interface.

5. Conclusions

This paper compares vocal one switch interfaces based on three different scanning algorithms with a traditional multi-word speech recognition-based interface. The aim is to provide a robust and efficient interface for virtual and augmented reality tasks to be performed hands-free.

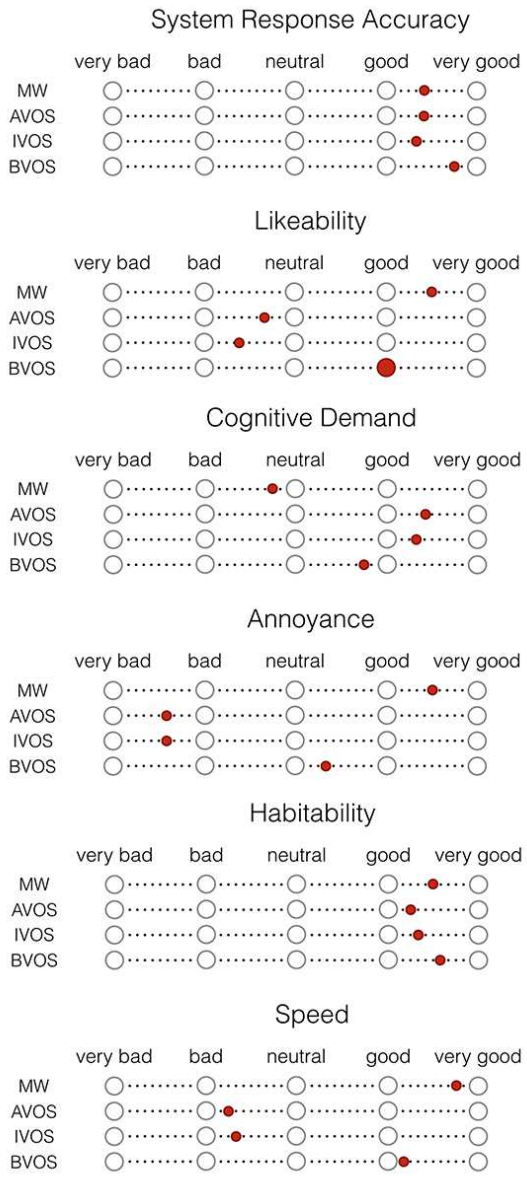

Figure 7: *Usability evaluation based on the SASSI usability principles.*

A bidirectional scanning algorithm has been added to the traditional automatic and inverse ones. The bidirectional algorithm is based on a three-state switch triggered by three words, thus enabling users to select available commands as if they were placed in a circular buffer. In this way, it is possible to improve robustness performance with respect to a multi-word solution with a limited overhead in completion times.

From the user's point of view, although it is the least in terms of robustness, the multi-word solution is preferred in seven cases of eleven; this is due to the fact that the users are asked for pronouncing a lower number of words and they accept a greater error ratio. On the other hand, the BVOS interface is perceived as the best alternative to the MW due to its improved robustness and slightly worse speed. AVOS and IVOS are ranked by the majority of the users as the worst solutions due to the huge gap in latency time with respect to the other two interfaces, although AVOS is the least prone to errors in noisy environments.

This study will be extended by considering also other forms of command activation beyond the vocal one; for instance, blinking detection promises to be a robust form of binary activation, which might completely overcome any problem related to noisy environments. Moreover, other scanning algorithms should be investigated when more complex applications have to be managed. The relationship among number of commands to be activated, robustness of the interface and latency time to reach the desired command is still an open problem.

References

[ALCDR15] ACED LÓPEZ S., CORNO F., DE RUSSIS L.: Playable one-switch video games for children with severe motor disabilities based on gnomon. In *7th International Conference on Intelligent Technologies for Interactive Entertainment, INTETAIN 2015, Torino, Italy, June 10-12, 2015* (2015), pp. 176–185. 2

[Ang92] ANGELO J.: Comparison of three computer scanning modes as an interface method for persons with cerebral palsy. *American Journal of Occupational Therapy 46*, 3 (1992), 217–222. doi:10.5014/ajot.46.3.217. 2

[BASQ*04] BLACKSTIEN-ADLER S., SHEIN F., QUINTAL J., BIRCH S., WEISS P. L. T.: Mouse manipulation through single-switch scanning. *Assistive Technology 16*, 1 (2004), 28–42. 2

[BCM09] BRODERICK T., CAMERON MACKAY D. J.: Fast and flexible selection with a single switch. *CoRR abs/0909.2450* (2009). 2, 6

[BKP01] BILLINGHURST M., KATO H., POUPYREV I.: The magicbook-moving seamlessly between reality and virtuality. *IEEE Computer Graphics and applications 21*, 3 (2001), 6–8. 2

[BKP08] BILLINGHURST M., KATO H., POUPYREV I.: Tangible augmented reality. *ACM SIGGRAPH ASIA 7* (2008). 2

[BM06] BARTIE P. J., MACKANESS W. A.: Development of a speech-based augmented reality system to support exploration of cityscape. *Transactions in GIS 10*, 1 (2006), 63–86. 2

[BR08] BISWAS P., ROBINSON P.: A new screen scanning system based on clustering screen objects. *Journal of Assistive Technologies 2*, 3 (2008), 24–31. 2

[BRC96] BURDEA G., RICHARD P., COIFFET P.: Multimodal virtual reality: Input-output devices, system integration, and human factors. *International Journal of Human-Computer Interaction 8*, 1 (1996), 5–24. 2

[DBH*09] DIERKER A., BOVERMANN T., HANHEIDE M., HERMANN T., SAGERER G.: A multimodal augmented reality system for alignment research. In *Proceedings of the 13th International Conference on Human-Computer Interaction* (2009), vol. 1, pp. 1–5. 2

[DBR13] DAM P., BRAZ P., RAPOSO A.: *A Study of Navigation and Selection Techniques in Virtual Environments Using Microsoft Kinect®*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 139–148. doi:10.1007/978-3-642-39405-8_17. 2

[GB10] GIBBONS C., BENETEAU E.: Functional performance using eye control and single switch scanning by people with als. *SIG 12 Perspectives on Augmentative and Alternative Communication 19*, 3 (2010), 64–69. 2

[GSZN03] GOOSE S., SUDARSKY S., ZHANG X., NAVAB N.: Speech-enabled augmented reality supporting mobile industrial maintenance. *IEEE Pervasive Computing 2*, 1 (2003), 65–70. 2

[GWZ97] GUPTA R., WHITNEY D., ZELTZER D.: Prototyping and design for assembly analysis using multimodal virtual environments. *Computer-Aided Design 29*, 8 (1997), 585–597. 2

[HBK07] HARDERS M., BIANCHI G., KNOERLEIN B.: Multimodal augmented reality in medicine. In *International Conference on Universal Access in Human-Computer Interaction* (2007), Springer, pp. 652–658. 2

[HG00] HONE K. S., GRAHAM R.: Towards a tool for the subjective assessment of speech system interfaces (sassi). *Natural Language Engineering 6*, 3&4 (2000), 287–303. 7

[HIII13] HASHIMOTO S., ISHIDA A., INAMI M., IGARASHI T.: Touchme: An augmented reality interface for remote robot control. *Journal of Robotics and Mechatronics 25*, 3 (6 2013), 529–537. 2

[KGMQ09] KIM J.-S., GRACANIN D., MATKOVIC K., QUEK F. K. H.: iphone/ipod touch as input devices for navigation in immersive virtual environments. In *VR* (2009), IEEE Computer Society, pp. 261–262. 2

[KVL07] KOK A. J., VAN LIERE R.: A multimodal virtual reality interface for 3d interaction with vtk. *Knowledge and Information Systems 13*, 2 (2007), 197–219. 2

[LLR*08] LÉCUYER A., LOTTE F., REILLY R. B., LEEB R., HIROSE M., SLATER M., ET AL.: Brain-computer interfaces, virtual reality, and videogames. *IEEE Computer 41*, 10 (2008), 66–72. 2

[LMP*16] LAMBERTI F., MANURI F., PARAVATI G., PIUMATTI G., SANNA A.: Using semantics to automatically generate speech interfaces for wearable virtual and augmented reality applications. *IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS PP*, 99 (2016), 1–13. doi:10.1109/THMS.2016.2573830. 2, 3

[LNBK04] LEE G. A., NELLES C., BILLINGHURST M., KIM G. J.: Immersive authoring of tangible augmented reality applications. In *Proceedings of the 3rd IEEE/ACM international Symposium on Mixed and Augmented Reality* (2004), IEEE Computer Society, pp. 172–181. 2

[MM09] MISTRY P., MAES P.: Sixthsense: A wearable gestural interface. In *ACM SIGGRAPH ASIA 2009 Sketches* (New York, NY, USA, 2009), SIGGRAPH ASIA '09, ACM, pp. 11:1–11:1. doi:10.1145/1667146.1667160. 2

[MMC09] MISTRY P., MAES P., CHANG L.: Wuw - wear ur world: A wearable gestural interface. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2009), CHI EA '09, ACM, pp. 4111–4116. doi:10.1145/1520340.1520626. 2

[MP15] MANURI F., PIUMATTI G.: A preliminary study of a hybrid user interface for augmented reality applications. In *7th International Conference on Intelligent Technologies for Interactive Entertainment, INTETAIN 2015, Torino, Italy, June 10-12, 2015* (2015), IEEE, pp. 37–41. 2

[Nav04] NAVARRO K. F.: Wearable, wireless brain computer interfaces in augmented reality environments. In *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on* (2004), vol. 2, IEEE, pp. 643–647. 2

[Nie96] NIELSEN J.: Usability metrics: Tracking interface improvements. *Ieee Software 13*, 6 (1996), 12. 6

[OBF03] OLWAL A., BENKO H., FEINER S.: Senseshapes: Using statistical geometry for object selection in a multimodal augmented reality system. In *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality* (2003), IEEE Computer Society, p. 300. 2

[OYT96] OHSHIMA T., YAMAMOTO H., TAMURA H.: Gaze-directed adaptive rendering for interacting with virtual space. In *Virtual reality annual international symposium, 1996., Proceedings of the IEEE 1996* (1996), IEEE, pp. 103–110. 2

[PLC08] PARK H. M., LEE S. H., CHOI J. S.: Wearable augmented reality system using gaze interaction. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality* (Washington, DC, USA, 2008), ISMAR '08, IEEE Computer Society, pp. 175–176. doi:10.1109/ISMAR.2008.4637353. 2

[Raj04] RAJAEE S.: Web browser software with single switch interface for pdas or computers. In *Bioengineering Conference, 2004. Proceedings of the IEEE 30th Annual Northeast* (2004), IEEE, pp. 253–254. 2

[RCH13] REGENBRECHT H., COLLINS J., HOERMANN S.: A leap-supported, hybrid ar interface approach. In *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration* (New York, NY, USA, 2013), OzCHI '13, ACM, pp. 281–284. doi:10.1145/2541016.2541053. 2

[SK99] SIMPSON R. C., KOESTER H. H.: Adaptive one-switch row-column scanning. *IEEE Trans Rehabil Eng 7*, 4 (Dec. 1999), 464–473. 2

[SLBM16] SANNA A., LAMBERTI F., BAZZANO F., MAGGIO L.: Developing touch-less interfaces to interact with 3d contents in public exhibitions. In *Third International Conference, AVR 2016, Lecce, Italy, June 15-18, 2016. Proceedings, Part II* (2016), LNCS Series, Springer, pp. 293–303. 2

[SMP*15] SANNA A., MANURI F., PIUMATTI G., PARAVATI G., LAMBERTI F., PEZZOLLA P.: A flexible ar-based training system for industrial maintenance. In *International Conference on Augmented and Virtual Reality* (2015), Springer, pp. 314–331. 3

[TT87] TREVIRANUS J., TANNOCK R.: A scanning computer access system for children with severe physical disabilities. *American Journal of Occupational Therapy 41*, 11 (1987), 733–738. 2

[Wil96] WILSON J.: *Virtual Reality for Industrial Application: Opportunities and Limitations*. Nottingham University Press, 1996. 2

[WYNC09] WAKITA Y., YAMANOBE N., NAGATA K., CLERC M.: Customize function of single switch user interface for robot arm to help a daily life. In *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on* (Feb 2009), pp. 294–299. 2

[YG98] YANCO H. A., GIPS J.: Driver performance using single switch scanning with a powered wheelchair: robotic assisted control versus traditional control. In *TITLE Proceedings of the RESNA'98 Annual Conference: The State of the Arts and Science (Minneapolis, Minnesota, June 26-30* (1998), ERIC, p. 309. 2