A Preliminary Study of a Hybrid User Interface for Augmented Reality Applications

Federico Manuri, Giovanni Piumatti
Politecnico di Torino
Dipartimento di Automatica e Informatica
C.so Duca degli Abruzzi 24,I-10129, Torino, Italy
Email: federico.manuri@polito.it, giovanni.piumatti@polito.it

***Abstract*—Augmented Reality (AR) applications are nowadays largely diffused in many fields of use, especially for entertainment, and the market of AR applications for mobile devices grows faster and faster. Moreover, new and innovative hardware for human-computer interaction has been deployed, such as the Leap Motion Controller. This paper presents some preliminary results in the design and development of a hybrid interface for hand-free augmented reality applications. The paper introduces a framework to interact with AR applications through a speech and gesture recognition-based interface. A Leap Motion Controller is mounted on top of AR glasses and a speech recognition module completes the system. Results have shown that, using the speech or the gesture recognition modules singularly, the robustness of the user interface is strongly dependent on environmental conditions. On the other hand, a combined usage of both modules can provide a more robust input.**

Keywords — Augmented Reality, hybrid user interface, human-computer interaction, speech recognition, gesture recognition.

I. Introduction

In the last years, Augmented Reality (AR) applications have become very popular, due to the diffusion of low-cost mobile devices such as smartphones and tablets. These devices are usually equipped with a camera and a GPS sensor, providing an ideal platform for delivering augmented reality contents. AR applications have been researched, investigated and developed in many fields: marketing [1]–[3], maintenance [4]–[7], tourism [8], [9] and most of all entertainment [10]–[13]. These applications will probably be even more widespread as new and more natural and comfortable AR devices, such as AR glasses [14] and lens [15], become available. Most of the AR applications developed for mobile devices, however, rely only on touch screen interaction and/or gyroscopic data. This poses the problem of how to interact intuitively with such applications while using AR glasses, without a physical interface. Recently, new and innovative hardware has surfaced, which allows users to interact naturally with computer devices, such as The Microsoft Kinect [16] or the Leap Motion Controller [17]. These devices, however, are usually used in desktop environments, due to their tethered nature. This paper proposes a novel user interface exploiting both gesture and speech recognition, on the assumption that the features offered by a device such as the Leap Motion Controller could become even more portable and reliable in the near future. The proposed interface allows users to interact with AR applications delivered through AR optical see-through glasses. The goal is to design a system that can "weight" the gestures and words recognized depending on the environmental conditions. Combining and evaluating the weight of the two inputs it is possible to produce a more robust input. The paper is organized as follows: Section 2 presents the state of the art of hands-free interaction interfaces, focusing on interfaces for AR applications. Section 3 outlines the proposed framework, describing both the hardware and software architecture. Preliminary tests and evaluation of the data acquired are discussed in Section 4. Section 5 presents an improved design of the decisional algorithm which takes into account both the modules input and the environmental conditions. Open problems and future developments are discussed in Section 6.

II. Background

In order to achieve hands-free interaction, the most used interfaces are speech-based. Speech recognition and understanding technologies have advanced greatly, but still suffer robustness issues [18]. One of the greatest difficulties is to correctly understand utterances with background noise, such as other people's chatter. For this reason, their use alone is not suited for a robust interface, as *false positive* detections might damage the user experience. Another possibility is represented by gesture-based interfaces. In order to achieve high accuracy, such interfaces use gloves or similar wearable components [19]. They allow to carry out complicated tasks, but are often uncomfortable and thus not suitable for a user interface targeted for entertainment applications. Using stereo cameras and computer vision software, it is possible to track bare hands [20], although to obtain a high precision they often require expensive equipment. By combining both speech and gesture recognition, some interfaces have overcome several of the limitations of the two single technologies, and they are focused mainly towards achieving natural interaction. A lot of attention has been given to manipulation tasks. In [21], a system was developed that allows users to grab, move and release virtual objects. The AR environment is shown in a separate display, and the scene is captured by an overhead camera. In similar projects, the display was substituted by Head-Mounted Displays (HMDs) or handheld devices [22] [23]. A Leap Motion Controller was attached to an HMD in order to track the hands of the user in [24]. Different kinds of

978-1-63190-061-7 © 2015 ICST
DOI 10.4108/icst.intetain.2015.259629

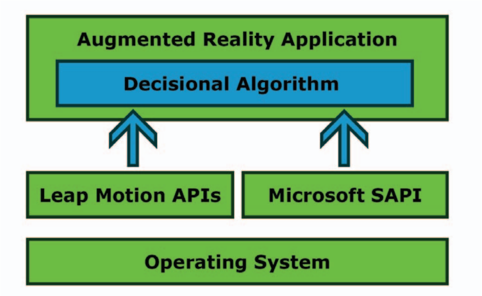

Fig. 1. Software Layers of the proposed framework

menus were developed in order to evaluate user acceptance. This paper proposes to integrate hand gesture and speech recognition to control the user interface , in order to achieve a better recognition performance and to avoid unintentional commands. The reliability of each module is evaluated and "weighted" to determine a more robust input, depending on the environmental conditions. The word *performance* is hereafter used with regard to the robustness of the system and it doesn't refer to usability or to real-time performance.

III. The Proposed Framework

The proposed framework consists of two different modules, one for gesture recognition and the other for speech recognition. Figure 1 shows the building layers for the software part of the framework. The AR application used for the preliminary tests was deployed on a laptop running the Windows 8 operating system. Through the Microsoft SAPI APIs [25], the words pronounced by the user are elaborated and sent to the application as control inputs. The Leap Motion APIs are used similarly for gesture recognition. To test the framework, the following hardware configuration was adopted: a Leap Motion Controller, mounted on top of Vuzix Star 1200XLD optical see-through AR glasses, and a Plantronic Discovery 975 Bluetooth headset. A specific support was designed and casted with a 3D printer to mount the Leap Motion on top of the Vuzix glasses. Figure 2 shows the Leap mounted on top of the glasses and the headset. The proposed AR glasses and the Leap Motion Controller were actually designed for desktop computers. However, there are tablets running the Windows 8 operating system and equipped with USB input and HDMI output that technically allow to deploy the proposed configuration on mobile devices. We are aware that more comfortable solutions may come up in the near future.

A. Gesture Recognition

The gesture recognition module should represent the main interface to control the application. The Leap Motion provides three different sets of information: the position of the hand in front of the Leap; information on the fingers, such as occlusion, in order to define and recognize poses; finally, when the user moves the hands, the Leap provides further information if a gesture is recognized. As a result, the first step is to define which set (or combination of sets) of gestures would be more reliable and natural when using the Leap in a vertical configuration.

The Leap APIs provide the automatic recognition of four different gestures:

1) the **Swipe** gesture, which consists in a swiping motion of the hand in front of the camera;
2) the **Circle** gesture is recognized when the tip of a finger draws a circle in front of the Leap;
3) the **KeyTap** gesture, which consists in rotating the tip of a finger down toward the palm and then springing it back to the original position;
4) the **ScreenTap** gesture, which consists in poking forward the tip of a finger and then springing it back to the original position.

All available gestures were tested to identify which ones perform better when the Leap is mounted on top of the AR glasses. More gestures were obtained by filtering the axis and the direction of the gesture. The Circle gesture was splitted into two different gestures, as it is possible to identify whether the gesture is being performed clockwise or counter-clockwise. Some restrictions were adopted to avoid *false positive*s, such as a minimum value for the Circle gesture radius. The Swipe along the z-axis was avoided because too similar to the gesture of positioning the hand in front of the Leap.

Moreover, poses were researched to evaluate another set of input information. Different poses were tested to find which could perform better, based on the number and type of fingers extended or occluded. The first issue that arised was that, if the user places the hand in front of the Leap and then moves the fingers to obtain the chosen pose, eventually gestures or other poses are recognized in the process. To fix the problem, a different approach was evaluated: first taking the pose, than moving the hand in front of the camera. This approach turned out to be more reliable but requires further investigation: the hand is not always correctly recognized if placed in front of the Leap not fully open. Finally, two different poses were defined to use in the test session:

1) the **Fist** pose, when all the five fingers are closed into a fist;
2) the **Victory** pose, when only the index and middle fingers are extended, forming a V shape.

These poses were tested to evaluate if they could perform better than gestures in some environmental conditions.

In the future, data about the hand position into 3D space will be evaluated as well. A virtual menu will be developed to provide proper feedback to the user, in order to obtain meaningful results during the test sessions.

B. Speech Recognition

Even if the proposed interface will mostly be controlled through hand gestures, speech input is nonetheless important. It allows to interact with the application when the hands are otherwise occupied and can express better some types of commands. The speech recognition module should be user-independent and capable of recognizing a moderate sized

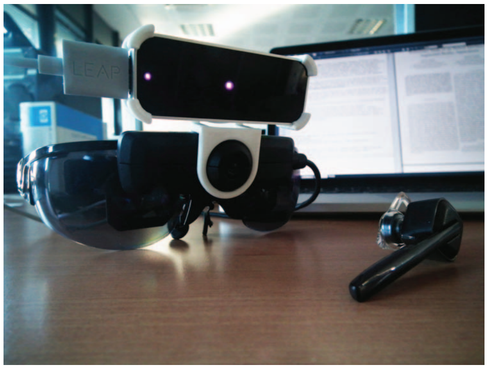

Fig. 2. Leap Motion Controller, Vuzix AR Glasses and Plantronic Headset

dictionary. Although in some cases convenient, continuous speech recognition has the drawback of generating a lot of *false positive* detections in noisy environments. As a solution, the present paper proposes to tie it to a specific gesture, in order to de/activate it at will. The interface features a few, key vocal commands, in order not to force the user to remember too many words and to improve the recognition rate. Moreover, the commands are context-dependent, in order to minimize the number of *false positive*s.

IV. Preliminary Tests

This section presents some preliminary results from testing both the gesture recognition and the speech recognition modules. The aim of these tests was to evaluate the reliability of the two modules singularly. The following step will be to combine the two modules to obtain a more robust system.

A. Gesture

Gesture recognition was tested using five different gestures:

1) **Left Swipe** gesture, a swipe from right to left.
2) **Right Swipe** gesture, a swipe from left to right.
3) **Circle gesture** clockwise.
4) **Circle gesture** counter-clockwise.
5) **Tap Gesture**, a swipe toward the camera.

Also the two poses Fist and Victory, described in the previous Section, were tested.

Table I shows the percentage of gestures and poses detected during the test. The tests were performed in three different light conditions: outdoor environment with direct sunlight, outdoor environment with diffused sunlight and indoor environment. A result is depicted as *true positive* if the gesture or pose was correctly recognized, *false negative* if the gesture or pose was not recognized or Incorrect Detection if the wrong gesture or pose was recognized.

A further test in outdoor environment with high light reflection was performed: in this case the behaviour of the Leap was so unpredictable that it was not possible to correctly collect data. It was not possible to distinguish between *false negative* and *incorrect detections* because of the high rate of *false positive*. In this context, by *false positive* the authors mean that the system recognized something even though no pose or gesture was performed.

The tests performed indoor are used as a reference because it is the most common environment for the Leap Motion Controller. For both gesture and pose recognition the *true positive* rate is over 75%. Pose recognition seems more robust than gesture recognition with a *true positive* rate of 90%. During the outdoor tests with diffuse sunlight the *false negative* rate is higher than the *incorrect detection* rate (exactly the opposite) compared to the previous test. In the last tests, performed in an environment with direct sunlight, the Leap could not recognize correctly almost any gesture. On the contrary, the pose recognition still performed well.

The tests reveal that environment with high light reflection should rely only on the speech recognition module as the Leap Controller becomes unpredictable. Moreover, the test show that using the Leap outdoor is still possible with reliable results.

B. Speech

Speech recognition was tested using the Microsoft Speech API (SAPI). The grammar used consisted of 7 separate commands, expressed either as a single word or a brief sentence (e.g. “play video”). Tests were carried out in three different environmental conditions: quiet, indistinct background noise and background chatter. Each command was uttered 10 times, for a total of 70 utterances. The recognized result is categorized as either *true positive* (correct detection), *false negative* (no detection) or *incorrect detection* (detection of the wrong word). Moreover, if a detection occurred without a command being uttered, this was categorized as *false positive*. Table II shows the percentage of words detected in the different environmental conditions, relative to the total number of words detected. The percentages corresponding to the uttered words alone (i.e. excluding *false positive*s) are shown in brackets.

Longer sentences appear to be more robust, as they are harder to be falsely recognized. On the other hand, if the utterance does not correspond exactly to the configured grammar, the sentence is not recognized. Single words are recognized correctly most of the time. If the grammar is designed carefully by avoiding similar sounding words, they are rarely misinterpreted. On the other hand, single words are also the ones more often falsely detected. Noisy conditions do not impact significantly the recognition rate, but they have a big effect on false detections. Indeed, *false positive*s were avoided only in a completely silent environment. Background noise and especially chatter increase significantly the false detection rate. Background chatter accounts for as many as 26% of the recognized words (which were never uttered), thus making speech recognition alone unreliable in noisy environments.

The test reveals also that background chatter is perceived by the microphone at about 48-58dB, whereas uttered commands are always greater than 60dB. This could provide a rough threshold to distinguish between utterances and noise, thus lowering the false detection rate. It should also be noted that if the recognition engine were to be trained to the speaker,

TABLE I
GESTURE RECOGNITION TEST RESULTS, 5 GESTURES REPEATED 10 TIMES EACH

	Gesture			Pose		
Environment (lux)	*true positive*	*false negative*	*incorrect detections*	*true positive*	*false negative*	*incorrect detections*
Indoor (400)	78%	8%	14%	90%	10%	0%
Outdoor, Diffused Sunlight (800)	78%	14%	8%	90%	10%	0%
Outdoor, Direct Sunlight (>1000)	2%	88%	10%	85%	15%	0%

TABLE II
SPEECH RECOGNITION TEST RESULTS

Environment (dB)	*true positive*	*false negative*	*incorrect detections*	*false positive*	*total words*
Quiet (40dB)	86% (86%)	12% (12%)	2% (2%)	0	70
Indistinct Noise (45-55dB)	88% (89%)	9% (9%)	1.5% (1.5%)	1.5%	71
Chatter (48-58dB)	64% (87%)	6% (8%)	4% (5%)	26%	95

recognition rate could rise even more. This was avoided to allow the interface to be used in more general contexts, but in some situations training could be considered.

C. Use case

In order to test a sample use case for the proposed system, a simple entertainment AR application was developed. The application leverages the concept of *magic book*, where the pages of a book are used to create augmented content. The application consists in just two pages, on which are displayed respectively a video and an animated 3D model. Interaction can take place either through speech or gesture recognition. The video and the animation can be played, paused and stopped. Moreover, the 3D model can be made bigger or smaller. The *play*, *pause* and *stop* commands are mapped with the corresponding words in case of speech recognition, and with the gestures **Victory** (for *play* and *pause*) and **Tap** (for *stop*) in case of gesture recognition. The *bigger* and *smaller* commands are again mapped to the same words for speech recognition and to **Swipe left** and **Swipe right**, respectively, for gesture recognition.

The application was tested by 7 students between 23 and 29 years of age. First both speech and gesture input methods were tested under neutral conditions (no interference). Speech recognition was by far the preferred method of interaction, due to the fact that it is simpler and does not require prior training. On the contrary, gestures were harder to learn and reproduce correctly. After that, the same actions were carried out in a noisy environment. As was expected, the speech recognition module became unusable at this point, due to the elevated number of *false positives*. The students were then able to deactivate the speech recognition module and interact through gestures alone.

Feedback from these tests revealed that interaction through speech was preferred because more natural and easy to learn. Gestures, instead, are less intuitive and require some prior training to get used to. Overall it was reported that the application's strongest feature was the fact that a fallback input method (gestures) was available, even if it is not as robust as the main method (speech).

V. DISCUSSION

The aim of this paper is to design a user interface that is as robust as possible in most contexts. The main objective is to virtually eliminate any *false positive* or *incorrect detections*, in order to avoid unintended commands to be executed. To this end, the authors propose to use data about the environmental conditions in order to better estimate the reliability of each module. The paper proposes to use a separate microphone and a light detector to determine how much these factors could influence the recognition results. Specifically, the microphone will detect the level of background noise, whereas the light detector will capture the amount of IR interference. The data from both sensors will then be normalized and used as a coefficient to adjust the corresponding recognition module's confidence value. Equations 1 and 2 illustrate this concept.

$$AC_s = C_s(1 - N(x)) \tag{1}$$

$$AC_g = C_g(1 - L(y)) \tag{2}$$

C_s and C_g represent the confidence respectively of speech and gesture recognition. Their values go from 0 (completely unreliable) to 1 (completely certain). $N(x)$ and $L(y)$ represent noise/interference distributions, respectively for sound and IR: x is the current noise level in decibel (dB) and y is the current IR level in lux. The resulting values AC_s and AC_g represent the adjusted confidence for speech and gesture recognition respectively. Furthermore, it could be useful to assign different weights to the different available commands, depending on their criticality. A highly critical command is one that, if executed, has "serious" consequences, and depends on the specific application. For example, a high-criticality command could be quitting a game without saving, whereas a low-criticality command could be displaying a picture. The more critical the command, the higher its weight, and vice versa. When a command is recognized either by the speech or by the gesture recognition module, its adjusted confidence value is calculated and compared with a threshold, considering the command's criticality as well. A command is accepted (i.e. executed) only if 3 is true.

$$AC_i \geq T_i^{min} + (T_i^{max} - T_i^{min})W_k \tag{3}$$

k represents the recognized command, whereas i is the input method (speech or gesture). T_i^{max} and T_i^{min} represent the maximum and minimum thresholds for input method i. W_k represents the weight (criticality) of command k, where a value of 0 means very low criticality and 1 indicates very high criticality.

In order to obtain satisfactory results, it is important that the $N(x)$ and $L(x)$ functions are set up correctly, as well as the values of T_i^{min} and T_i^{max} for both recognition modules. $N(x)$ and $L(x)$ are almost certainly non-decreasing, and are very likely to be linear, logarithmic or exponential. Various tests will be carried out in order to establish the best distribution.

The T_i^{min} and T_i^{max} thresholds depend both on the underlying recognition technology and the specific application. Some tests will be carried out in order to establish meaningful reference values. T_i^{max} becomes significant for more critical commands, therefore it should be tuned accordingly. Specifically, in an environment with high levels of interference, T_i^{max} controls the rejection rate of false positives, and should be set in order to achieve the lowest acceptable false positive rate. At the same time, it should allow explicit commands to be recognized easily in an interference-free environment.

The value of T_i^{min} can be more flexible, as it controls the acceptance of low-criticality commands. It should be tuned in order to achieve results at least comparable with those obtained by using the corresponding recognition module on its own. Increasing it would lower the acceptance rate of false detections, but at the same time it would also lower the chance of accepting intentional commands.

VI. Conclusions and Future Work

This paper presents a preliminary study of a hybrid interface for AR applications. Gesture recognition with the Leap Motion and speech recognition through Microsoft SAPI were investigated to evaluate how to design a more robust interface that can make use of the best of both technologies. Although the proposed framework is still in an early stage, the potential of the proposed interface is notable and could be of great advantage when using wearable AR devices. The next step of experimentation will be focused on developing and testing the proposed hybrid system. If the system does not prove robust enough, further research will be aimed to enhance the performance of the speech and gesture modules. An option could be to develop custom gesture recognition algorithms and to add a module for hand tracking through an RGB camera as well. For the speech recognition module, it could be possible to set up a decibel threshold to ignore sounds under a certain value and lower the false detection rate. Moreover, the overall performance of the system will be evaluated, considering not only the robustness but also the usability, especially in applications that require real-time interaction.

Acknowledgment

This work is funded by the EASE-R^3 project: Integrated framework for a cost-effective and ease of Repair, Renovation and Re-use of machine tools within modern factory, FP7, FoF.NMP.2013-8, Grant agreement no: 608771.

References

[1] M. Bulearca and D. Tamarjan, *Augmented Reality: A Sustainable Marketing Tool?*, Global Business and Management Research: An International Journal, vol. 2, pages 237-252, 2010

[2] Ikea AR Catalogue, website http://www.ikea.com/ca/en/about_ikea/newsitem/2014catalogue

[3] Converse Shoe Sampler applications, website http://www.rga.com/work/converse-the-sampler-3/

[4] S.J. Henderson and S. Feiner, *Exploring the Benefits of Augmented Reality Documentation for Maintenance and Repair*, IEEE Trans. on Visualization and Computer Graphics, vol. 17, pages 1355-1368, 2011

[5] G. Terenzi and G. Basile, *Smart Maintenance: An Augmented Reality Platform for Training and Fields Operations in the Manufacturing Industry*, ARMEDIA Augmented Reality Blog, 2014,

[6] F. Lamberti, F. Manuri, A. Sanna, G. Paravati, P. Pezzolla and P. Montuschi, *Challenges, opportunities and future trends of emerging techniques for Augmented Reality-based maintenance*, IEEE Transactions On Emerging Topics In Computing, IEEE, in press

[7] F. Manuri, A. Sanna, F. Lamberti, G. Paravati and P. Pezzolla, *A workflow analysis for implementing AR-based maintenance procedures*, Proceedings 1st International Conference on Augmented and Virtual Reality, in press

[8] Z. Yovcheva, D. Buhalis and C. Gatzidis, *Smartphone Augmented Reality Applications for Tourism*, e-Review of Tourism Research (eRTR), vol. 10, pages 63-66, 2012

[9] GeoTravel and Places applications' website http://www.augmentedworks.com/

[10] W. Piekarski and B. Thomas, *ARQuake: the outdoor augmented reality gaming system*, Communications of the ACM journal, vol. 45, pages 36-38, ACM, 2002

[11] Ingress application's website: https://www.ingress.com/

[12] Theodolite application's website: http://hunter.pairsite.com/theodolite/

[13] Drakerz Confrontation application's website: http://www.drakerz.com/

[14] The Google Glass project web site: http://www.google.com/glass/start/

[15] The Innovega web site: http://innovega-inc.com/

[16] Microsoft Kinect's website: http://www.microsoft.com/en-us/kinectforwindows/

[17] Leap Motion's website: https://www.leapmotion.com/

[18] S. Varges and M. Purver, *Robust language analysis and generation for spoken dialogue systems*, Proceedings of the ECAI workshop on Development and Evaluation of Robust Spoken Dialogue Systems for Real Applications, Riva del Garda, Italy, 2006

[19] J. Y. Lee, G. W. Rhee and D. W. Seo, *Hand gesture-based tangible interactions for manipulating virtual objects in a mixed reality environment*, The International Journal of Advanced Manufacturing Technology, vol. 51, pages 1069-1082, Springer, 2010

[20] O. Hilliges, D. Kim, S. Izadi, M. Weiss and A. Wilson, *HoloDesk: direct 3d interactions with a situated see-through display*, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 2421-2430, ACM, 2012

[21] M. Lee, M. Billinghurst, W. Baek, R. Green and W. Woo, *A usability study of multimodal input in an augmented reality environment*, Journal on Virtual Reality, vol. 17, number 4, pages 293-305, Springer, 2013

[22] G. Park, T. Ha and W. Woo, *Hand Tracking with a Near-Range Depth Camera for Virtual Object Manipulation in an Wearable Augmented Reality*, Virtual, Augmented and Mixed Reality. Designing and Developing Virtual and Augmented Environments, pages 396-405, Springer, 2014

[23] T. Piumsomboon, A. Clark and M. Billinghurst, *[DEMO] G-SIAR: Gesture-speech interface for augmented reality*, Mixed and Augmented Reality (ISMAR), 2014 IEEE International Symposium on, pages 365-366, IEEE, 2014

[24] Z. He and X. Yang, *Hand-based interaction for object manipulation with augmented reality glasses*, Proceedings of the 13th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry, pages 227-230, ACM, 2014

[25] Microsoft SAPI website: https://msdn.microsoft.com/en-us/library/ee125663\%28v=vs.85\%29.aspx