Helping deaf and hard-of-hearing people by combining augmented reality and speech technologies

M R Mirzaei, S Ghorshi, M Mortazavi

School of Science and Engineering, Sharif University of Technology-International Campus
Kish Island, IRAN

mrmirzaie@gmail.com, ghorshi@sharif.edu, mortazavi@sharif.edu

ABSTRACT

Recently, many studies have shown that the Augmented Reality (AR), Automatic Speech Recognition (ASR) and Text-to-Speech Synthesis (TTS) can be used to help people with disabilities. In this paper, we combine these technologies to make a new system, called "ASRAR", for helping deaf people. This system can take a narrator's speech and convert it into a readable text, and show the text directly on AR displays. Since most deaf people are unable to make meaningful sounds, we use a TTS system to make the system more usable for them. The results of testing the system show that its accuracy is over 85 percent, using different ASR engines, in different places. The results of testing TTS engines show that the processing time is less than three seconds and the spelling of correct words is 90 percent. Moreover, the result of a survey shows that more than 80 percent of deaf people are very interested in using the ASRAR system for communication.

1. INTRODUCTION

Today, using new technologies to help people with disabilities is highly regarded and much research in this area is underway. The important issue is how to combine and use these technologies together to make them more applicable. A number of recent studies have shown that AR and VR can be used to help people with disabilities (Lange, et al., 2010) (Zainuddin and Zaman, 2009). AR gives a possibility to disabled people to control and manage the information and adapt it easily to a desired form to improve their interactions with people (Zayed and Sharawy, 2010) (Passig and Eden, 2000). Another technology that can be used to help disabled people is Automatic Speech Recognition (ASR). ASR technology gives the possibility to disabled people to control and use computers by voice commands, e.g., to control robotic arms (Mosbah, 2006). In addition, ASR allows deaf or hard-of-hearing people to participate in conference rooms (Mosbah, 2006). Text-to-Speech Synthesis (TTS) is one of the important systems that is used to help disabled people (Handley, 2009). TTS converts the display information, such as text files or web pages, into the speech for the visually challenged people, such as blind or deaf people (Dutoit, 1997) (Lopez-Ludena, et al., 2011).

Communication with people is a basic need for everyone, but some people are not able to communicate well due to some disabilities. A group of these people is deaf people, and communication is the main problem among them. Deaf people communicate visually and physically rather than audibly. Therefore, they have some problems in their relationship with people. Usually, deaf people use sign-language to communicate with each other, but people have no desire to learn sign-language. For this reason, many people feel awkward or become frustrated trying to communicate with deaf people, especially when no interpreter is available. Besides this problem, most deaf people are unable to make meaningful sounds and also have problems in visual literacy, such as participate in university classes or scientific meetings. Therefore, we tried to find a way to solve the communication problem among deaf and hard-of-hearing people, using new technologies. In this paper, we present a new system, called "ASRAR", for combining AR, ASR and TTS technologies. This is a new system with multiple features, but helping deaf people to communicate with ordinary people is its main goal. Our proposed system uses the narrator's speech to make the speech visible to deaf people on AR display. This system helps deaf people to see what the narrator says, and also the narrator does not need to learn sign-language to communicate with deaf people. Furthermore, deaf people can talk to the narrator, using a computer. The rest of this paper is organized as follows. In Section 2, related work in AR, ASR and TTS area is presented. In Section 3, we explain our proposed system in terms of the structure, system design and system requirements. In Section 4, the experimental results of testing the system are presented, and finally in Section 5, we summarize and conclude the paper.

©2012 ICDVRAT; ISBN 978-0-7049-1545-9

2. RELATED WORK

Recently, much research has been carried out in AR, ASR and TTS technologies. Some studies have shown that these technologies can be used to help people with disabilities. In the AR field, Zainuddin et al. used AR to make an AR-Book called the "Augmented Reality Book Science in Deaf (AR-SiD)," which contains 3D modeling objects, with using markers and sign-language symbols, to improve the learning process of deaf students (Zainuddin, et al., 2010). Also, in the ASR and TTS fields, Lopez-Ludena et al. developed an advanced speech communication system for deaf people with visual user interface, 3D avatar module and TTS engine to show the effects of using ASR and TTS technologies to help people with disabilities (Lopez-Ludena, et al., 2011). Some researchers have worked on using AR and ASR technologies together. Irawati et al. used a combination of AR and ASR technologies in AR design applications and showed the possibility of using the speech to arrange the virtual objects in virtual environments (Irawati, et al., 2006). Hanlon et al., in a similar work, used the speech interfaces as an attractive solution to the problem of using keyboard and mouse in AR design applications (Hanlon, et al., 2009). Goose et al. sowed that AR and ASR can also be used in AR industrial maintenance applications. They made a multimodal AR interface framework in the SEAR (Speech-Enabled AR) project with a context-sensitive speech dialogue for maintenance technicians to check the factory components (Goose, et al., 2003).

Compared to our work, explained systems are used by speakers, and the speech is used by the system to specify a command to arrange virtual objects in AR environments, or it is used by the system as a parameter to identify objects without using input devices. Furthermore, all the above systems have used AR markers to detect and show virtual objects in an AR environment. Our system captures the narrator's speech and uses the ASR engine to convert the speech to text, so that the AR engine shows the text directly on AR display. In addition, the main user of the ASRAR system is a deaf person that does not talk in the system's scenario, and could use the TTS engine to talk to the narrator. Moreover, the ASRAR is a marker less AR application and uses the narrator's facial expressions as a marker to show the virtual objects in an AR environment.

3. THE PROPOSED SYSTEM

Our proposed system, called "ASRAR", includes a variety of technologies. It consists of two main parts: hardware and software. In hardware part, some hardware requirements, such as camera, microphone, speaker and display, are required, and in software part, we develop the core of the system that consists of the AR, ASR and TTS engines, the Joiner Algorithm and the Auto-Save script. All these parts can be brought together in an integrated system. Figure 1 shows the overall view of the ASRAR with an Ultra Mobile PC (UMPC). Deaf people can also use AR Head Mounted Displays (AR-HMD) or mobile phones to see AR environments. Since the ASRAR is developed as a cross-platform application, it can be used in many portable devices.

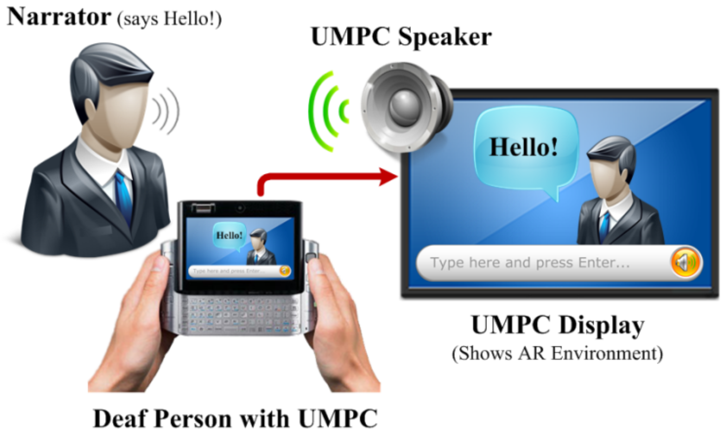

Figure 1. *Overall view of the ASRAR system.*

In ASRAR system's scenario, the ASR engine collects the speech from the detected narrator, and the AR engine realizes the scenario. The Joiner Algorithm is used to combine AR and ASR engines to work together. To achieve some goals, it is required to use some automated process scripts that will be integrated to the system in the future. The system uses the ASR engine to recognize the narrator's speech and convert the speech to text. The TTS engine is used by the system to convert the input text into the speech for communication proposes between the deaf person and the narrator. The system also uses the AR engine to display the text as a dynamic object in an AR environment.

To get the video and the speech of the narrator, the ASRAR uses built-in cameras and microphones on UMPC (or mobile phone), or AR-HMD, and to show the objects in an AR environment, it uses UMPC or AR-HMD's display. In addition, the UMPC's keyboard or a virtual keyboard on tablets is used by deaf people to write a text. Then, the TTS engine will convert the text into the speech, and the speech will be played by the speakers. A deaf person can use our system in any place without carrying AR markers because of the following important features in the system.

1. The face detection techniques are used instead of markers to detect the narrator.
2. The TTS engine is used by deaf people to convert the text into the speech.

3.1 System Structure

In this section, we explain the structure of the system and the components that is used in it. In our system's scenario, a deaf person focuses the camera to the narrator. The camera can be a web camera connected to or embedded in a computer or a mobile phone, or mounted on an AR-HMD. The camera captures the video and the built-in or external microphone, captures the narrator's speech, and the speaker plays the text for the narrator. Figure 2 shows the structure of the ASRAR system.

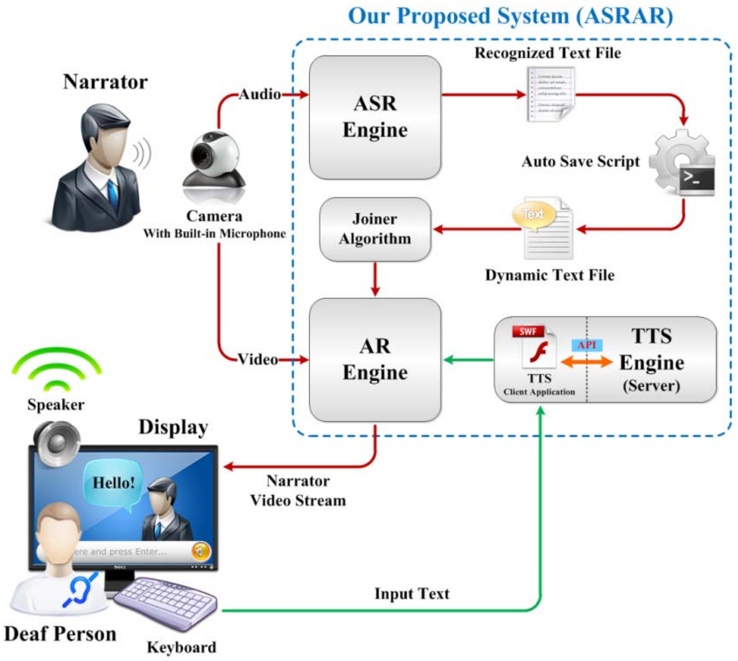

Figure 2. *The ASRAR system structure.*

3.1.1 The ASR Engine. To show the power of the ASRAR and due to experimental work, Dragon Naturally Speaking, Dragon Dictate (Nuance Communications Inc., 2011) and Microsoft Speech Recognition (Microsoft Corp., 2011) are used as powerful external ASR engines. The accuracy of ASR engines depends on their processing and databases, and is usually measured with Word Error Rate (WER) (Mihelic and Zibert, 2008) (Kalra, et al., 2010). The output of ASR engines is the recognized text strings that are written in a text file by the engines. The ASRAR uses the dynamic version of this text file as an input to the Joiner Algorithm. Using the ASR engines, the ASRAR system captures the text strings and writes them respectively in a text file as its text database. For initial testing of the ASRAR, a .TXT file is used by the system to save recognized text strings. On the other hand, the ASRAR can be developed with an internal temporary text array to save the recognized text strings.

Every time the narrator says a word, the ASR engine captures the word in a text file, but the word is not saved in the text file by the engine. The Joiner Algorithm needs the updated version of the text file, which means every word must be saved in the text file. We propose the Auto-Save script to do the saving operation automatically every 1 second when a word is written in the text files by the engine. This script provides the updated version of the text file as "Dynamic Text File." Then, the text file is used by the Joiner Algorithm as the input text, which is shown in Figure 3. The Auto-Save script is a small automated process script that is written with VBScript in Windows Operating System (OS) and with AppleScript in Mac OS. Moreover, it can be developed with other programming languages or as internal coding in the system. To avoid increasing size of the text file with useless text strings, the ASRAR is programmed to clean the text file completely, every time it is run by the user.

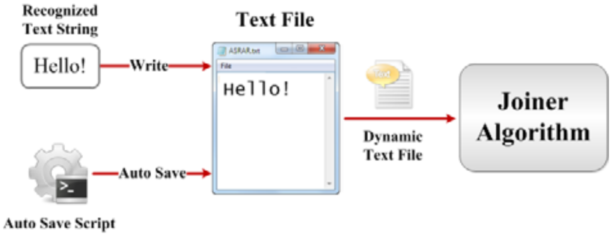

Figure 3. *The Auto-Save script working process.*

3.1.2 The Joiner Algorithm. To make relations between the AR and ASR engines, we propose a module that is called "Joiner Algorithm." The Joiner Algorithm module is a Flash application with AS3 coding (Braunstein, et al., 2007) to load dynamic text strings on a speech bubble image or a subtitle line. The output of the Joiner Algorithm is a .SWF application that is displayed by the AR engine in an AR environment, Figure 4.

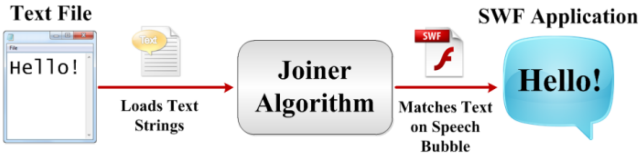

Figure 4. *The Joiner Algorithm working process.*

The Joiner Algorithm is developed to load only the last word in the text file. This development is optional and developers can change it easily to show more words (like a sentence). With this development, the Joiner Algorithm is always looking for the last word in the text file and when it finds a new word, it loads the word on the speech bubble image. In the second development, we change the Joiner Algorithm to load more words (like a sentence) in different styles, such as a subtitle line.

3.1.3 The AR Engine. In the AR engine, the FLARManager (Transmote, 2011) and OpenCV (Open Computer Vision Library by Intel, 2011) (Bradski and Kaehler, 2008) are used. The FLARToolKit library (Spark Project Team, 2011) is also used in the FLARManager framework to make flash supported AR applications, since it is compatible with many 3D frameworks (Hohl, 2008), such as Papervision3D (Hello Enjoy Company, 2011). The FLARManager also supports multiple marker detection (Cawood and Falia, 2008) (Arusoaie, et al., 2010) that will enable us to develop the system in the future for detecting more than one narrator in the environment. The purpose of using these toolkits is for their availability to export the system as .SWF, hence allow the availability of making the system cross-platform. Our system uses the face detection instead of the marker. If the marker had been used in the ASRAR, we should have designed specific markers, and the users should have carried the marker. Therefore, the ASRAR is developed as a marker-less AR application. In this case, some marker detection challenges, such as marker choice and marker creation that makes the system more complex, are avoided. We found a ported version of OpenCV for AS3 called "Marilena Object Detection" (Spark Project Team, 2011) for facial recognition, object detection and motion tracking. It has the best performance among AS3 libraries for face detection, and doesn't have a heavy process in the AR engine. The Marilena library is combined to the FLARToolkit library to develop the AR engine with Adobe Flash Builder IDE (Adobe System Inc., 2011). The block diagram of AR engine is shown in Figure 5.

In the AR engine, after the narrator's video frames are captured by the camera, the face detection module identifies the face of the current narrator, and sends this information to the Position and Orientation module. For initial testing of the ASRAR, the face detection module is developed to detect and use only the face of one person as the narrator's face. In this method, the ASRAR knows where the narrator's face is located in the video frames. Simultaneously, the Joiner Algorithm sends its output, which is a .SWF file, to the Position and Orientation module. Then, the Position and Orientation module places the speech bubble .SWF file to the narrator's face information, and sends it to the Render module. Finally, the Render module augments the output of the Position and Orientation module on AR display, near the narrator's face. It also renders the TTS client application at a fixed position in AR environments. Figure 6 shows the narrator's speech and the TTS client application that are visible in an AR environment.

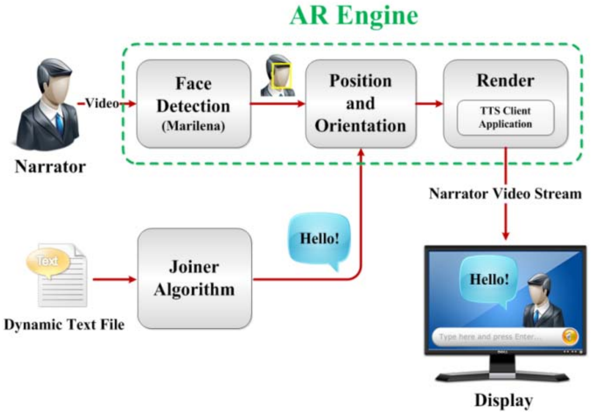

Figure 5. *Block diagram of the AR engine.*

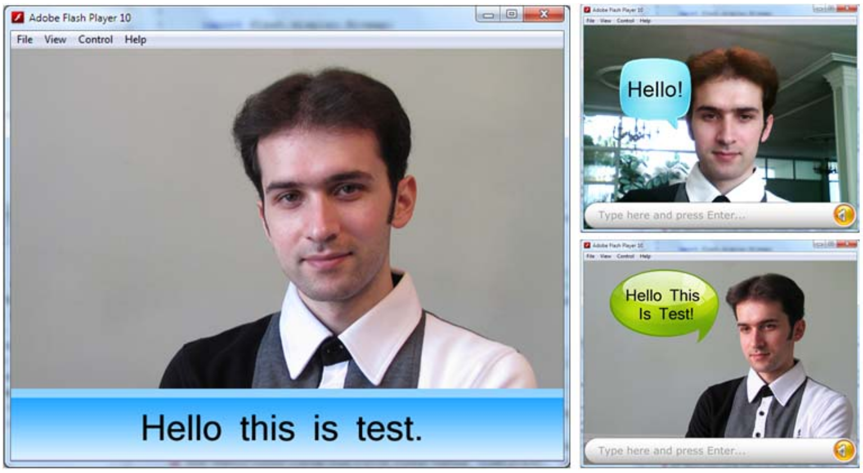

Figure 6. *Different text style's view in the ASRAR system.*

The speech bubble, which is accommodated near the narrator's face, is sensitive to the narrator movements and when the narrator moves, the speech bubble also moves smoothly. In addition, it is zoomed in or out without any delay, when the narrator moves forward or backward. It is possible to make the speech bubble sensitive to the narrator's face rotations, but we disable this feature by default, since the text is faded and become unreadable with rotating the speech bubble. This problem will be fixed if we use a subtitle style. For the initial testing of the ASRAR, the user cannot select the text style's appearance in AR environments.

3.1.4 The TTS Engine. In the ASRAR system, the TTS engine has two main parts: the TTS client application and the TTS engine server. The TTS client application is a small Flash application that is connected to the TTS engine server by APIs, and is added to AR environments by the AR engine. Figure 7 illustrates the TTS client application structure. For initial testing of the TTS engine, the user must connect the system to the internet to be able to connect the TTS engine client application to the TTS engine server, using the TTS engine's APIs.

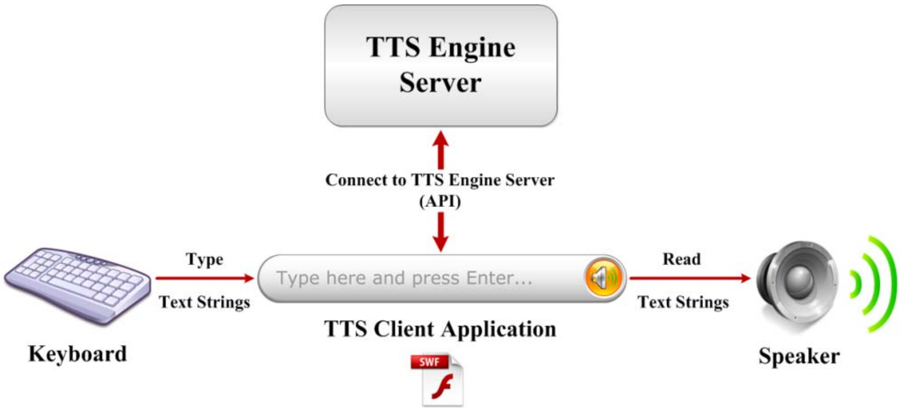

Figure 7. *The TTS client application working process.*

When the narrator says something, the deaf person can write a text and reply using the UMPC's keyboard or virtual keyboard on tablets. This text will be played by the speakers when the Enter key is pressed by the deaf person. This conversation is removed automatically by the system to make the system ready for the next conversation.

3.2 Hardware Requirements

To run the ASRAR in the basic state, it is not necessary to have very powerful hardware specifications. Generally, we need some specific hardware requirements to run the ASRAR in a portable device with Windows OS. These hardware requirements are shown in Table 1.

Table 1. *Hardware Requirements.*

Device	Description
Processor	Intel Mobile or Core Due 1.0 GHz.
Memory	1 GB.
Camera	640*480 VGA sensor or more, compatible with USB 2.0 or 1.1, or built-in cameras.
Microphone	All types of built-in or external microphone.

The power of ASR engines is different in portable devices and depends on hardware specifications (Mihelic and Zibert, 2008). Nevertheless, the ASRAR is run easily on new mobile phones, due to recent advances in mobile hardware. For initial testing of the ASRAR in Windows and Mac OSs, we use an UMPC and a MacBook Air with specific hardware, such as Intel processors, 4GB memory, and 5mp web camera with noise-cancelling microphone. The ASRAR and also the external ASR engines work fine with this particular hardware and provide very good results.

4. EVALUATION RESULTS

We evaluated and tested the ASRAR in four different noisy environments, using three popular and powerful ASR engines. In addition, two online powerful TTS engine servers, such as the Google (Google Android Developers, 2011) and the demo version of AT&T (AT&T, 2011), are used to test the TTS engine. We classified our tests according to different conditions that a deaf person might be. It is tried to choose a condition that reflects the performance of the ASRAR in terms of word error rate and recognition accuracy. To get better results, an external microphone with a noise-cancelling feature and a powerful digital camera are used to capture the narrator's video. Also, it is assumed that the distance between the narrator and the camera is only 1 meter, and the narrator speaks slowly with a clear English accent.

4.1 Classification of the System Tests

To test the ASRAR, the tests are classified to: Test 1 for ideal ASR engine, Test 2 for ideal narrator, Test 3 for ideal environment and Test 4 for crowded environment. In Test 1, we used a writer script, which was developed with VBScript programming language, instead of the ASR engine to write some words directly into the system's text file at the specific times. Therefore, this test did not have any phonetic problem and noise in the system. We did this particular test to know the effects of working the ASR engine in 100 percent performance on the system. For initial testing of the ASRAR, the writer script was developed to write some words automatically every two seconds in the text file. This script worked like a narrator says words every two seconds, and the ASR engine writes these words in the text file.

In Test 2, we used the transcription feature in the ASR engines to transcribe a clear recorded man's audio file into the ASRAR's text file. Transcription is the ability of ASR engine to get the audio file of the speech and convert the audio file to text strings (Mihelic and Zibert, 2008). The transcription process needs to open and read recorded audio file to perform, so there are some delay in it. However, this delay is not important in the ASRAR because it is due to some processes in ASR engine. Of course in a real environment we do not use a recorded speech audio file and transcription process. Nevertheless, we will see acceptable results on average if the transcription feature is used. Dragon Naturally Speaking and Dragon Dictate have the ability to open recorded audio files and transcribe them to text strings, but Microsoft Speech Recognition does not have this ability in built-in free version on Windows OS. For transcribe our audio file with Microsoft Speech Recognition, we used internal playback of the audio file with the audio player software, and then Microsoft Speech Recognition converted the audio file to text strings.

In Test 3, we tested the ASRAR in a closed environment with very little noise, such as a home room or a personal office, and it is assumed that the narrator speaks slowly and loudly with clear English accent. In addition, to get better results, we assumed that the distance between the narrator and the camera (with microphone) is only 1 meter. Finally, In Test 4, we tested the ASRAR in a crowded environment with much noise to obtain the better performance results. In this test, the ASRAR was used in an outdoor environment with more noise.

4.1.1 Comparison of the Tests Result. We tested and compared the results of the four different tests with three popular and powerful ASR engines, which were the Dragon Naturally Speaking, the Dragon Dictate, and the Microsoft Speech Recognition. Each of these ASR engines has features that make them different from each other. The ASRAR is a cross-platform system, so we tested it in different operating systems by using different external ASR engines. The Dragon Naturally Speaking is a very powerful ASR engine that gives more features rather than other ASR engines. It provides impressive speech recognition and transcription abilities, and is surprisingly quick and accurate. The Microsoft Speech Recognition is a built-in speech recognition tool that the Microsoft included it free in Windows Vista and Windows 7 OSs. The features and accuracy of the Microsoft Speech Recognition certainly make it more useful. The Dragon Dictate is another speech recognition software that developed by Nuance Communications for using in Apple Macintosh OS. The result of our tests shows the performance of the ASR engines in different conditions, and it also reflects the performance of using the system in different places with very good approximation. For each of the tests, we considered the different environmental conditions, but these conditions were identical to each ASR engine. It is very important for us to know whether or not the system can work in different environments in terms of performance and recognition accuracy. The results in Figure 8 show that the recognition accuracy of the Dragon Naturally Speaking is 90 percent on average compared to other ASR engines. These results also show that the accuracy of the system is over 85 percent on average, using different ASR engines in different noisy environments. Therefore, Figure 8 shows that this system can be very useful for helping deaf people in different noisy places.

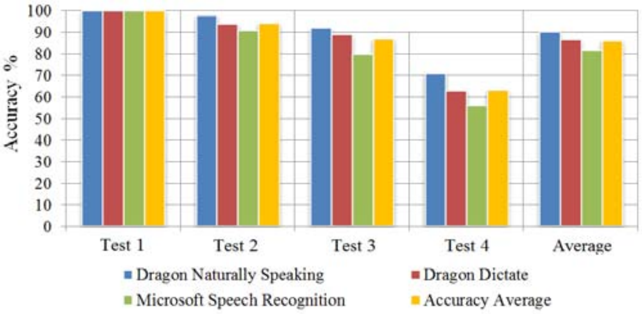

Figure 8. *Comparison results of testing the ASRAR system in different environments.*

4.2 Testing the TTS Engine

To test the TTS engine, two online powerful TTS engine servers, such as the AT&T (demo version) and the Google, are used. The Google TTS engine server contains only a female speaker, whereas the AT&T database consists of male and female speakers with different English accents, such as British and American. We evaluated the results for 10 random words, in terms of engine processing time, voice quality and percentage of spelling correct words in unpredictable sentences. Figure 9 illustrates the results of the AT&T and the Google TTS engine servers. Figure 9 (a) shows that the Google and the AT&T on-line TTS engine servers have very fast response time, and the average processing time is less than three seconds using 128 Kbps ADSL internet connection that is a reasonable result for on-line TTS engines. Of course, the speed of typing words by a deaf person is important, but we assumed that this time is two seconds for each word. Figure 9 (b) shows the voice quality of the Google and the AT&T TTS engine servers. In this test, numbers zero to 10 are used to rank voice quality, where number 10 shows that the speech is completely recognizable, and number zero shows that the speech is unrecognizable. It is noted that the voice quality is above 8, which is very close to natural speaking. Figure 9 (c) also shows the percentage of spelling correct word in unpredictable sentences. It is clear from Figure 9 (c) that the average spelling correct word is above 90 percent that is very reliable for powerful TTS engines.

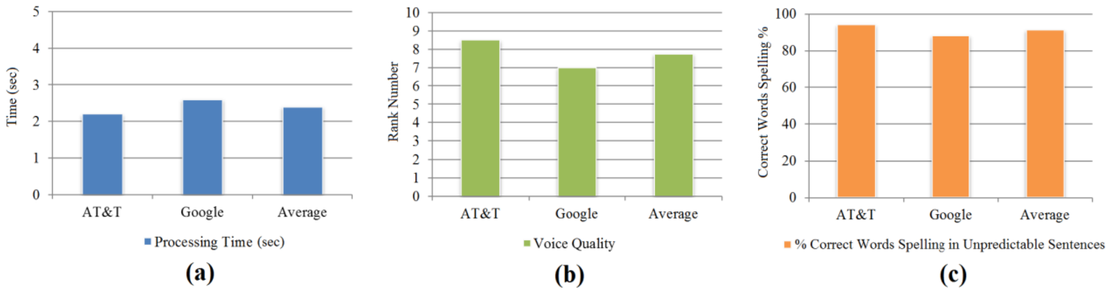

Figure 9. *The results of testing TTS engines in the ASRAR system.*

4.3 Processing Time of the System

It is assumed that the system works in real-time, which means every time the text file is changed by the ASR engine the results appear immediately on AR display. Thus, the processing time of word recognition and displaying it on AR display depend directly on the power of the ASR engine and hardware specifications. Figure 10 shows the processing time of the system in four different steps. In each step, a random word was captured and recognized by the ASR engine.

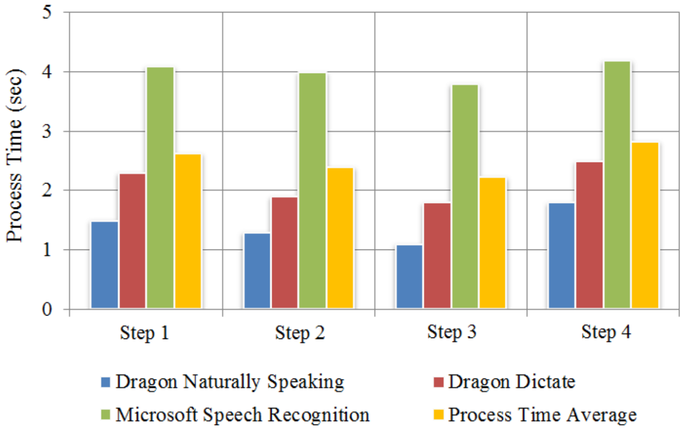

Figure 10. *The processing time of the ASR engines, with average.*

It can be noted from Figure 10 that the average processing time of the different ASR engines is less than three seconds that is not good enough, but it is reasonable for initial testing of the system. The processing time will reduce if we use the more powerful ASR engines.

4.4 Survey about the System

In this paper, we conducted a survey among 100 deaf people and 100 ordinary people to understand the interest rate of using different communication methods between them. The following question helped us to clarify our objectives for the survey: "Will you intend to use such our system for communication in the future?" The communication methods in this survey are assigned as Text, Sign-language and the ASRAR (AR+ASR). Since we did not have the necessary hardware (such as AR-HMD) to implement the system, people were not being able to test the ASRAR. Therefore, the survey was only conducted as questionnaires and people could choose different answers simultaneously. In addition, we provided a manual file of the system to make deaf people familiar with the ASRAR system, which contains the ASRAR structure and working process with images of the system. The result of this survey is presented in Figure 11.

The results in Figure 11 show that more than 90 percent of both groups are interested in using the ASRAR system to communicate with each other, instead of using only text or just sign-language. It is noted from Figure 11 that more than 80 percent of deaf people and less than 30 percent of ordinary people prefer to use sign-language. Also, these results show that deaf people and ordinary people have problems to communicate with each other, using only sign-language. Therefore, the ASRAR can be useful to solve the communication problem between deaf and ordinary people.

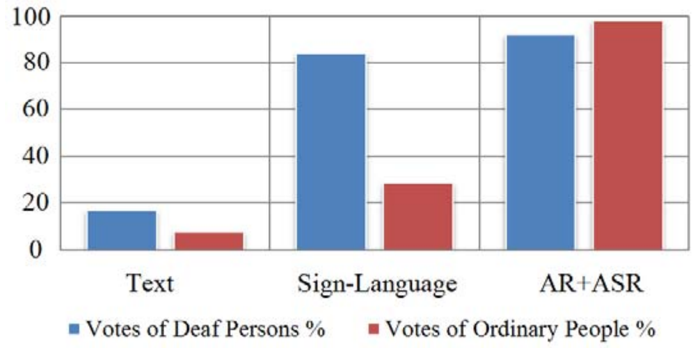

Figure 11. *Interest rate of three separate communication methods between deaf and ordinary people.*

5. CONCLUSIONS

This paper proposed a new system, called "ASRAR", to help deaf people to communicate with people and vice versa, by finding a common factor between AR and ASR technology, which is the text string. The common factor helped us to combine AR and ASR technologies. Our proposed system makes the speech visible to deaf people on AR display. The comparison of the ASRAR system's accuracy in different tests showed that the system is consistent and provides acceptable results with current new ASR engines. The results of testing the ASRAR in the different environments showed that this system acts very well in many situations that a deaf person might be and provides acceptable results with current ASR engines.

The results of processing time of the ASRAR indicated that the average processing time of word recognition and displaying it on AR display is less than three seconds, using today ASR engines, which is reasonable for these engines. Moreover, two powerful TTS engines were added to the system to convert the text into the speech. The results of testing online TTS engine servers, such as the Google and the demo version of AT&T, showed that the response time and the average processing time is less than three seconds. The results also showed that the voice quality is very close to natural speaking.

A survey was also conducted to know the usability of this particular system between deaf people. The results of the survey showed that almost all deaf people will use our proposed system as an assistant to communicate with ordinary people in the future. In this paper, we showed that AR, ASR and TTS technologies have a high potential to combine and advance. Therefore, these technologies are in a position to grow and offer new possibilities to the world of technology. Hopefully, our proposed system will be an alternative tool for deaf people to improve their communication skills.

6. REFERENCES

A Arusoaie, A I Cristei, M A Livadariu, V Manea, and A Iftene (2010), Augmented Reality, *Proc. of the 12th IEEE Intl. Symposium on Symbolic and Numeric Algorithms for Scientific Computing,* pp. 502–509.

Adobe Systems Inc. (2011), *Adobe Flash Builder,* http://www.adobe.com/products/flash-builder.html, Accessed 10 June 2011.

A Kalra, S Singh, and S Singh (2010), Speech Recognition, *The Intl. Journal of Computer Science and Network Security (IJCSNS),* **10**, *6*, pp. 216–221.

AT&T (2011), *AT&T Natural Voices,* http://www.naturalvoices.att.com, Accessed 20 September 2011.

B B Mosbah (2006), Speech Recognition for Disabilities People, *Proc. of the 2nd Information and Communication Technologies (ICTTA),* Syria, pp. 864–869.

B S Lange, P Requejo, S M Flynn, A A Rizzo, F J Cuevas, L Baker, and C Winstein (2010), The Potential of Virtual Reality and Gaming to Assist Successful Aging with Disability, *Journal of Physical Medicine and Rehabilitation Clinics of North America,* **21**, *2*, pp. 339–356.

D Passig and S Eden (2000), Improving Flexible Thinking in Deaf and hard of hearing children with Virtual Reality Technology, *American Annuals of Deaf,* **145**, *3*, pp. 286–291.

E Kaiser, A Olwal, D McGee, H Benko, A Corradini, X Li, P Cohen, and S Feiner (2003), Mutual Disambiguation of 3D Multimodal Interaction in Augmented and Virtual Reality, *Proc. of the 5th Intl. Conf. on Multimodal Interfaces,* Vancouver, BC, Canada, ACM Press, pp. 12–19.

F Mihelic and J Zibert (2008), *Speech Recognition, Technologies and Applications*, InTech Open Access Publisher.

G Bradski and A Kaehler (2008), *Learning OpenCV: Computer Vision with the OpenCV Library*, O'Reilly Media.

Google Android Developers (2011), *Using Text-to-Speech*, http://developer.android.com/resources/articles/tts.html, Accessed 20 September 2011.

Hello Enjoy Company (2011), *Papervision 3D*, http://blog.papervision3d.org/, Accessed 8 May 2011.

H S Zayed and M I Sharawy (2010), ARSC: An Augmented Reality Solution for the Education Field, *The International Journal of Computer & Education*, **56**, *4*, pp. 1045–1061.

Microsoft Corp. (2011), *Microsoft Speech Recognition*, http://www.microsoft.com/en-us/tellme/, Accessed 10 June 2011.

N Hanlon, B M Namee, and J D Kelleher (2009), Just Say It: An Evaluation of Speech Interfaces for Augmented Reality Design Applications, *Proc. of the 20th Irish Conf. on Artificial and Cognitive Science (AICS)*, pp. 134–143.

N M Zainuddin and H B Zaman (2009), Augmented Reality in Science Education for Deaf Students: Preliminary Analysis, *Presented at Regional Conf. on Special Needs Education*, Faculty of Education, Malaya University.

N M M Zainuddin, H B Zaman, and A Ahmad (2010), Developing Augmented Reality Book for Deaf in Science: The Determining Factors, *Proc. of the IEEE Intl. Symposium in Information Technology (ITSim)*, pp. 1–4.

Nuance Communications Inc. (2011), *Dragon Speech Recognition Software*, http://nuance.com/dragon/index.htm, Accessed 14 June 2011.

Open Computer Vision Library (2011), *OpenCV*, http://sourceforge.net/projects/opencvlibrary/, Accessed 12 May 2011.

R Braunstein, M H Wright, and J J Noble (2007), *ActionScript 3.0 Bible*, Wiley.

R Kheir and T Way (2007), Inclusion of Deaf Students in Computer Science Classes Using Real-Time Speech Transcription, *Proc. of the 12th ACM Annual SIGCSE Conf. on Innovation and Technology in Computer Science Education (ITiCSE)*, USA, pp. 261–265.

S Cawood and M Falia (2008), *Augmented Reality: A Practical Guide*, Pragmatic Bookshelf.

S Goose, S Sudarsky, X Zhang, and N Navab (2003), Speech-Enabled Augmented Reality Supporting Mobile Industrial Maintenance, *The Journal of Pervasive Computing*, **2**, *1*, pp. 65–70.

S Irawati, S Green, M Billinghurst, A Duenser, and H Ko (2006), Move the Couch Where?: Developing an Augmented Reality Multimodal Interface, *Proc. of 5th IEEE and ACM Intl. Symposium on Mixed and Augmented Reality*, pp. 183–186.

Spark Project Team (2011), *Marilena Face Detection*, http://www.libspark.org/wiki/mash/Marilena, Accessed 14 June 2011.

Spark Project Team (2011), *FLARToolKit*, http://www.libspark.org/wiki/saqoosha/FLARToolKit/en, Accessed 8 May 2011.

T Dutoit (1997), *An Introduction to Text-to-Speech Synthesis*, Kluwer Academic Publishers.

Transmote (2011), *FLARManager: Augmented Reality in Flash*, http://words.transmote.com/wp/flarmanager/, Accessed 8 May 2011.

V Lopez-Ludena, R San-Segundo, R Martin, D Sanchez, and A Garcia (2011), Evaluating a Speech Communication System for Deaf People, *IEEE Latin America Transactions*, **9**, *4*, pp. 556–570.

W R Sherman and A B Craig (2003), *Understanding Virtual Reality*, Morgan Kaufmann Publisher.

W Hohl (2008), *Interactive Environment with Open-Source Software: 3D Walkthrough and Augmented Reality for Architects with Blender 2.43, DART 3.0 and ARToolkit 2.72*, Springer Vienna Architecture.

Z Handley (2009), Is Text-to-Speech Synthesis Ready for Use in Computer-Assisted Language Learning? *The International Journal of Speech Communication*, **51**, *10*, pp. 906–919.