An Avatar Based Translation System from Arabic Speech to Arabic Sign Language for Deaf People

Sami M Halawani, Daut Daman, Sarudin Kari, Ab Rahman Ahmad,

Faculty of Computing and Information Technology – Rabigh
King Abdulaziz University, Rabigh 21911, Kingdom of Saudi Arabia

Summary
The sign language is the primary means of communication among the deaf community and normal people. It is the natural language for the community, using hand movements and other gestures. Many attempts are being made to develop an Arabic Sign Language (ARSL) and distribute it among members of the deaf community and those concerned. Building an accurate sign language recognition system is of a great importance in order to efficiently facilitating this communication. In this research, an Avatar Based Translation System for ArSL (ABTS–ArSL) is developed using Microsoft Window 7 Speech Recognition Engine. A proper training on the Speech Recognition tutorial is suggested prior to using ABTS–ArSL to achieve better voice recognition. ABTS–ArSL grammar is developed and embedded in Microsoft Visual Basic 2008. The Arabic alphabets and single numbers with their respective signs and avatar images are stored in different directories and invoked upon receiving a word uttered by a user. ABTS–ArSL is a visual natural language model that might be used for communication by both groups in Arabic speaking countries.

Key words:
Deaf community, Arabic Sign Language, Arabic Sign Language Grammar, Avatar Based Translation System, Speech Recognition

1. Introduction

The world population [1] has surpassed 7 Billion in 2013 where nearly 400 million people live in Middle East and North Africa. The Arabic language forms the unifying feature of the region and it is the fifth most spoken language [2] in the world. Though different areas use local varieties of Arabic, all shares in the use of the modern standardized language derived from the language of Al-Quran. It is used in education, workplaces, government and also the media. Modern Standard Arabic largely follows the grammatical standards and uses much of the same vocabulary. The 70 million deaf people worldwide [3] use two distinct different varieties of a language in different social situations, such as a formal at work and an informal at home by a sign language. The sign language is the primary means of communication among them. Arabic Sign Language (*ArSL*) is required to be developed that will have a significant impact to the deaf community in Saudi Arabia and in general, the Arab world.

American Sign Language (*ASL*) [4] is the sign language used for deaf communities in the United States and also English-speaking community in Canada. *ASL* signs have phonemic components, including movement of the face, torso and also the hands. The word models for *ASL* developed in [5] transfer between different signers and different aspects. The advantage is one could use large amounts of avatar data in combination with a smaller amount of human data to spot a large number of words in human data. Transfer learning is made possible by representing blocks of video with intermediate discriminative features which based on splits of the data. The same splits construction in avatar and human data and proper clustering, yielding the features discriminative and semantically similar, similar signers feature imply similar words. The transfer learning is demonstrated in two scenarios, namely, from avatar to a frontally viewed human signer and from an avatar to human signer in a three-quarter view.

British Sign Language (*BSL*) [6] used in the United Kingdom (*UK*) is the preferred language of some deaf people in the *UK*. The language uses space and involves movement of the hands, body, face and head. *TESSA* in [7] is an experimental system using the appropriate sequence of signs in *BSL*. It aims to aid transactions between a deaf person and a clerk in a post office by translating the clerk's speech to sign language. A speech recognizer recognizes speech from the clerk and the system then synthesizes using a specially-developed avatar. The evaluation undertaken by deaf users and Post office clerks are discussed the findings to be used in the development of an improved system. Both *ASL* and *BSL* have their own perceptions and applications to the local needs but with similarity that exists, both sign languages may be adopted to *ArSL*.

Most Arabic-speaking countries [8] use many and different sign languages, however, the sign used for alphabets are similar. There are attempts are being made to develop a standard *ARSL* and distribute it among members of the deaf community and those concerned. The Arabic language is diglossic, but *ARSL* is not. The interaction

Manuscript received December 5, 2013
Manuscript revised December 20, 2013

between a deaf community and a hearing one might be minimal and basically concerned around families with deaf members, relatives of the deaf, and sometimes play friends and professionals. To date, *ARSL* is ongoing developmental phases and in recent years there have been an awareness of the existence of communities consisting of people with these disabilities.

ARSL [8] developed as independent systems of communication among Arab community and they are not interpretations of standard Arabic or from any spoken mother tongue. Finger spelling, for example, is not used to read out or communicate the standard form of Arabic. It is used to spell out proper nouns and words that do not have sign correspondence. *ArSL* is not particularly different from other known sign languages, such as *ASL* and *BSL*. It is basically manual languages made from phonemes that involve configuration of hands – hand shape, placement/space – position of hand in relation to body, and movement – directions/contacts within space. In addition, *ARSL* makes use of other features like the face, mouth, and tongue. *ArSL* also exhibits similar forms to other established sign languages, such as links between form and meaning in graphics form. Fig. 1 shows some examples of sign language [8] in pictorial form.

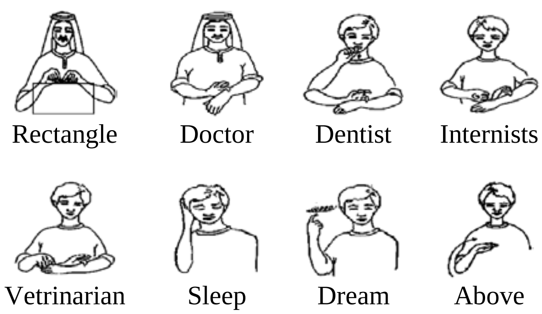


Fig. 1 Some examples of Sign Language

To date the research on *ArSL* are growing fast. *ArSL* Translation System on mobile devices was introduced by [9]. The software translates texts into sign language animations to improve deaf lives especially in communication and accessing information. The common applications on mobile phones are voice transmission and data transmission. The Wireless Application Protocol is an open, global specification that empowers mobile users with wireless devices to easily access and interact with information and services instantly.

The research in [10] presented an automatic translation system for gestures of manual alphabets in the *ArSL*. The proposed *ArSL* Alphabets Translator (*ArSLAT*) system does not rely on using any gloves or visual markings for the recognition job. It dealt with bare-hand images which allow users to interact with *ArSLAT* in a natural way. Experiments showed that *ArSLAT* recognized the Arabic alphabets with greater accuracy using Minimum Distance Classifier over Multilayer Perceptron classifier.

A contributing factor in [11] is the difficulty with which deaf children are acquiring learning concepts early in life. The discussion is on a highly interactive software using avatars (3D Character Module) to process and translate free Arabic input to *ArSL*. Such natural language is based on morphological, and partially syntactic and semantic analysis. The designed prototype is for teaching Arabic numerals, basic mathematical operations, some basic words and Arabic text translation to *ArSL*, developed and tested on deaf students.

An Arabic version of Automated Speech Recognition System (*ASR*) developed by [12] is based on the open source Sphinx-4, from the Carnegie Mellon University (*CMU*). *CMU* Sphinx is a large-vocabulary; speaker-independent, continuous speech recognition system based on a discrete Hidden Markov Model (*HMM*). The model built using utilities from the OpenSource *CMU* Sphinx to demonstrate the possibility of adapting *ASR* to Arabic voice recognition.

The obvious issue raised in [13] is that *ArSL* do not get as much focus as other sign languages, such as *ASL*. The software package provides a complete communication system for deaf to ensure communication between them and normal people as well as among themselves. It consists of sign language translation and chat tools in assisting the deaf to easily communicate with their friends, completing their computerized tasks and become more confident with Arabic language.

The use of polynomial classifiers [14] as a classification engine is discussed for the recognition of *ArSL* alphabets. Based on these classifiers, *ArSL* system is built and measured its performance using real *ArSL* data collected from deaf people. The system provides superior recognition results when compared with previously published results using *ANFIS*-based classification on the same dataset and feature extraction methodology. The comparison shown in terms of the number of misclassified test patterns yielded a very significant reduction in the rate of misclassifications on the training data and on the test data.

An image based system for *ArSL* recognition is proposed by [15]. A Gaussian skin color model is used to detect the centroid of signer's face. This reference is to track the hands movement using region growing from the sequence of images comprising the signs. A number of features selected from the detected hand regions across the sequence of images are recognized and performed using a *HMM*. The results achieved the outstanding recognition accuracies for a vast selected data set of signs with leave one out method.

Sign language recognition is a promising application that breaks the barrier between the deaf and normal people. It is the natural language for the deaf community, using hand movements and other gestures. Many ongoing researches stressed on some theoretical issues supported by some methods in assisting the deaf community to communicate better using *ArSL*. The outcome of this research is an Avatar Based Translation System for *ArSL* (*ABTS–ArSL*) grammar developed to focus on practical sign languages. The details are discussed in the following section.

2. Materials and Methods

2.1 The Arabic Alphabets and Numbers

The Arabic alphabets are written, read from right to left and horizontally. It has no capital letters. These 28 alphabets and the single numbers from 0 to 9 are tabulated in Table 1. The 29th alphabet, ء (Hamza) is not part of the alphabet series as it is used with أ (Alif).

Table 1: The 28 Arabic Alphabets and Numbers 0 to 9

Arabic	ر	ذ	د	خ	ح	ج	ث	ت	ب	أ
English	*R*	TH	*D*	KH	*H*	*J*	TH	*T*	*B*	*A*
Arabic	ف	غ	ع	ظ	ط	ض	ص	ش	س	ز
English	*F*	GH	'	TH	*T*	*D*	*S*	SH	*S*	*Z*
Arabic			ي	و	ه	ن	م	ل	ك	ق
English			*Y*	*W*	*H*	*N*	*M*	*L*	*K*	*Q*
Arabic Numbers	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

From Table 1 and [16], it can be seen some of the sounds in Arabic are also in English, and vice-versa. For example, the Arabic alphabet ب can be pronounced as the alphabet *B* in English whereas the Arabic ز sounds just like the z in English. The Arabic versions of *K* (ك), *M* (م), *N* (ن), *F* (ف), and *J* (ج) in [17] are all just the same. The alphabet ل is not exactly the same sound as the English *L*. The Arabic ل is pronounced with the tip of the tongue touching the roof of the mouth a bit farther back. The alphabet ر is a "rolled *R*" is completely different from English *R*. More in [17], the Arabic sounds ث also exist in English as "*TH*" in *thin* or *through*. The alphabet ذ makes the sound "*TH*" as in *them* or *this*. The alphabet ش makes the sound "*SH*" as in *shy* or *she*. The alphabet ي acts just like a *Y*. It sound like a long vowel at the end of a word such as "*EE*" in *meet* or it can be a consonant *Y*. The alphabet Hamza (ء) represents the glottal stop. It is pronounced by stopping the flow of breath at the back of the mouth cavity (the glottis) and it is also written as a diacritic.

The alphabets ح, خ, ص, ض, ط, ظ, ع, and غ have a unique way to pronounce and don't exist in English alphabets. The different tones [18] in pronunciation are between the alphabets ت and ط; د, ض and ث; ذ and ظ; أ and ع; ك and ق; غ and خ; ح and ه. For the alphabet ث, the *TH* sounds like the word "thing", where as *TH* used for the alphabet ذ, the pronunciation may be considered as the word "that". The alphabet ق needs much deeper tone compared to the alphabet *Q*. The alphabet ه is a hot *H* (glottal consonant) and sounds like when eating hot and spicy food with throat is like on fire.

Learning and differentiating those sounds are necessary to have a good Arabic accent. The readiness to expose as much to the sounds and listen how the Arabic is spoken make anyone able to speak correctly.

Table 2 outlines the suggested words phoneme mentioned in [10,14,19] to represent Arabic alphabets and Numbers (0 to 9) respectively.

Table 2: The words phonemes for Arabic Alphabets / Numbers (0 to 9)

Alphabet	ج	ث	ت	ب	أ
[10]	Jiem	Tha	Ta	Ba	Alef
[14]	Jim	Tha	Ta	Ba	Alif
Alphabet	ر	ذ	د	خ	ح
[10]	Ra	Thal	Dal	Kha	Ha
[14]	Ra	Thal	Dal	Kha	Ha
Alphabet	ض	ص	ش	س	ز
[10]	Dhad	Sad	Shien	Sien	Zay
[14]	Dhad	Sad	Shin	Sin	Za
Alphabet	ف	غ	ع	ظ	ط
[10]	Fa	Ghayn	Ayn	Thah	Tah
[14]	Fa	Ghayn	Ayn	Zad	Tad
Alphabet	ن	م	ل	ك	ق
[10]	Noon	Miem	Lam	Kaf	Qaf
[14]	Nun	Mim	Lam	Kaf	Qaf
Alphabet		ي	و	ه	
[10]		Ya	Waw	He	
[14]		Ya	Waw	He	
Numbers	**4**	**3**	**2**	**1**	**0**
[19]	Arba'a	Thalatha	Ithnan	Wahid	Sifr
Numbers	**9**	**8**	**7**	**6**	**5**
[19]	Tis'a	Thamaniya	Sab'a	Sitta	Khamsa

2.2 The Alphabets, Numbers, Sign and Avatar Data

Fig. 2 and 3 show the respective signs and avatars for 28 Alphabet and numbers (0 to 9). These are in the form of hand display, the placement in relation to body, the movement and its contact within space. The representation might be captured in the form of 3*D* images and videos.

In this research, an Avatar is simply the graphical representation of the word that will be translated to *ArSL*. Hence, the Avatar database is developed to contain and represent the 28 Arabic alphabets and numbers. Other than these manual shapes, the non manual features like those of the face, mouth and tongue are also can be used. The images of 28 alphabets and numbers (0 to 9), their corresponding signs and avatars are kept in the different directories which can be easily referred to. All images are represented in the JPG format for easy access.

Fig. 2 *ArSL* for 28 Alphabets signs and avatars

2.3 The speech recognition process

Generally, speech recognition process [3] contains three steps, namely, processing the speech which is in acoustic, extracting feature and recognizing the speech. The process is based on the model as shown in Fig. 4.

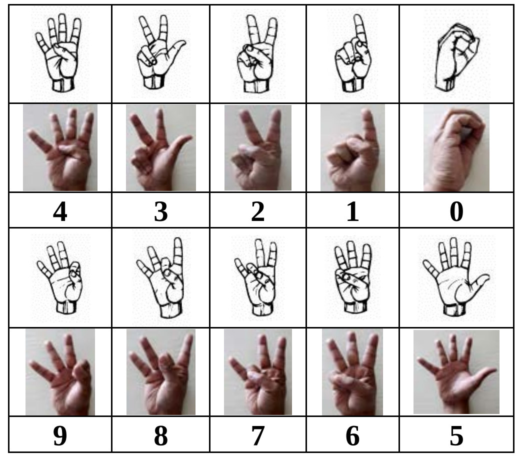


Fig. 3 *ArSL* for Numbers (0 to 9) signs and avatars

The first process is to digitize the speech that will be recognized. The second step is to extract the feature from a digital signal in time domain then feed to Linear Predictive Coding (*LPC*) spectrum analysis for extracting the signal or called as frame normalizing. *LPC* is used to extract the *LPC* coefficients from the speech tokens. *LPC* coefficients are the converted to Cepstral coefficients and the latter are normalized in between +1 and -1. The Cepstral coefficients are served as input to the neural networks.

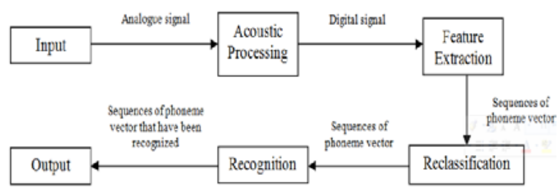


Fig. 4 The Speech Recognition Process

In this research, the speech recognition provided by Windows 7 in [20] is applied since a user's voice can be recognized automatically by the system. A user is required to perform the following training in order to acquire the maximum possible recognized speech. In the system, the voice can be used to control the computer by simply saying a command that the computer will respond to and also dictating text to the computer. More information can be found on the topics "What can I do with Speech Recognition?".

A microphone should be connected to a computer prior to the execution of following steps to setting up Speech Recognition in Windows 7.

Set up a microphone

- Click the Start button, Control Panel, Ease of Access, and then Speech Recognition to open Speech Recognition.
- Click Set up Microphone.
- Follow the instructions on the screen.

The success of speech recognition is directly depended on the quality of the microphone and the common types of microphones used are headset and desktop microphones. Headset microphones are considered better since they are less prone to picking up extraneous sounds.
A speech training tutorial in Windows 7 will assist a user to exploit the commands used with Speech Recognition and the tutorial takes about 30 minutes to complete. Fig. 5 and 6 showed the respective speech recognition main menu and pop-up windows display upon executing some items in the main menu. The following steps must be executed in order to run the speech training tutorial.

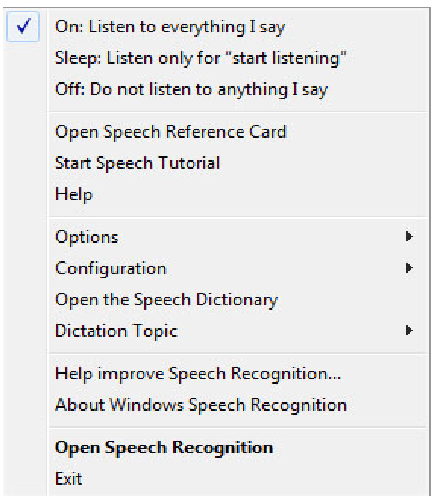


Fig. 5 The Speech Recognition Main Menu

The following steps must be executed in order to run the speech training tutorial.

Teach a user how to talk to a computer

- Click the Start button, Control Panel, Ease of Access, and then Speech Recognition to open Speech Recognition.
- Click Take Speech Tutorial.
- Follow the instructions in the Speech Recognition tutorial.

Train a computer to recognize a speech

There is a unique voice profile used by Speech Recognition to recognize user's voice and spoken commands. Once Speech Recognition is utilized, a user's voice profile gets more detailed and consequently improving computer's ability to understand his/her voice. The following the steps have to be executed in order to train a computer to recognize a speech.

- Click the Start button, Control Panel, Ease of Access, and then Speech Recognition to open Speech Recognition.
- Click Train your computer to better understand you.
- Follow the instructions on the screen.

Fig. 6 The Speech Recognition pop-up Window Messages

The implementation of speech recognition

The implementation in [21] written for C# provides an overview and examples in Windows Forms application by the following operations:

1. Initialize the speech recognizer.
   To create an instance of the shared recognizer in Windows, use the *SpeechRecognizer* class.
   SpeechRecognizer SR = new SpeechRecognizer();
2. Create a speech recognition grammar.
   The grammar is created by using the constructors and methods on the *GrammarBuilder* and *Choices* classes. For a simple grammar that recognizes the words "blue", "green" or "red", these words are added in a *Choices* object using the *Add* method. Then the *GrammarBuilder* instance is created and using the *Append* method to insert the colors object in *that* instance. Finally, the *GrammarBuilder* instance is the initialized. It should be noted that, a user must speak exactly one of the elements added by the *Choices* instance in order to attain the match between user speech and the grammar.
3. Load the grammar into the speech recognizer.
   The grammar created in the previous operation must be loaded and passed into the speech recognizer by calling the *LoadGrammar(Grammar)* method.
4. Register for speech recognition event notification.
   The speech recognizer raises a number of the *SpeechRecognized* events during its operation, when it accepts a user utterance with a grammar. A notification of this event is registered by appending an *EventHandler* instance via
   SR_SpeechRecognized,
   the name written by a developer.
5. Create a handler for the speech recognition event.
   A handler created for the *SpeechRecognized* event displays the text of the recognized word or phrase using the *Result* property on the
   SpeechRecognizedEventArgs parameter, e.

3. Results and Discussion

3.1 Setting words phoneme for the Arabic Alphabets and Numbers

Tables 3 and 4 showed the set of words phoneme for the 28 alphabets and numbers (0 to 9) based on [22] Arabic pronunciation.

Table 3: The words phoneme for the Alphabets

Alphabet	ج	ث	ت	ب	أ
[22]	جيم	ثا	تا	با	الف
Words	Jeem	Thaa	Taa	Baa	Alif
Alphabet	ر	ذ	د	خ	ح
[22]	را	ذال	دال	خا	حا
Words	Raa	Thaal	Daal	Khaa	Haa
Alphabet	ض	ص	ش	س	ز
[22]	ضاد	صاد	شين	سين	زا
Words	Dahd	Sahd	Sheen	Seen	Zaa
Alphabet	ف	غ	ع	ظ	ط
[22]	فا	غين	عين	ظا	طا
Words	Faa	Ghayn	Ayn	Thaa	Tta
Alphabet	ن	م	ل	ك	ق
[22]	نون	ميم	لام	كاف	قاف
Words	Nuu n	Meem	Laam	Kaaf	Qaaf
Alphabet		ي	و	ه	
[22]		يا	واو	ها	
Words		Yaa	Waaw	Ha	

Table 4: The words phoneme for the Numbers (0 to 9)

Numbers	**4**	**3**	**2**	**1**	**0**
[22]	أربعة	ثلاثة	إثنان	واحد	صفر
Words	Arbaa	Thalatha	Ithnai n	Wahi d	Sifer
Numbers	**9**	**8**	**7**	**6**	**5**
[22]	تسعة	ثمانية	سبعة	ستة	خمسة
Words	Tisaa	Thamaniy a	Sabaa	Sitta	Khamsa

3.2 Creating speech recognition grammar

The Speech Recognition grammars provided by Windows 7 are available only in English, French, Spanish, German, Japanese, Simplified Chinese, and Traditional Chinese. However, the Arabic script has been adopted for use in a wide variety of languages. Such adaptations may feature altered or new characters to represent phonemes that do not appear in Arabic phonology.

The challenge is to build the language grammar that allocates suitable word phonemes written with the Arabic script for a user's voice to be recognized. With a proper training, however, the implementation of a grammar in other languages may become feasible. Fig. 7 shows the architecture of *ABTS–ArSL*.

In this research, a speech recognition application in [21,23] is adapted to perform the following operations and based on Microsoft Visual Basic 2008 programming (*MS-VB*2008) structure.

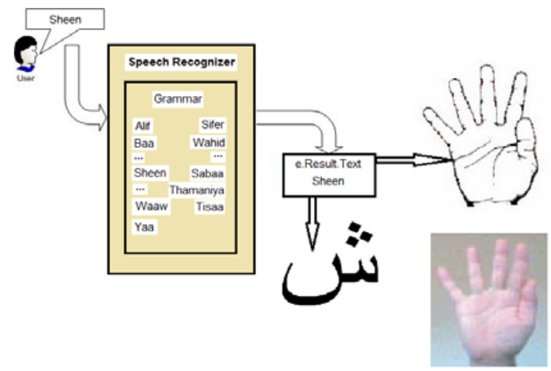


Fig. 7 The architecture of *ABTS–ArSL*

1. The component for system speech recognition is provided by *MS-VB*2008 and must be added as reference in MS-VB2008 and in the programming source code as
   ```
   Imports System.Speech
   ```
2. The definition for *ABTS–ArSL* as a new Speech Recognition is defined as follows
   ```
   Dim WithEvents ABTS As New
   Recognition.SpeechRecognizer
   ```
3. The definitions for the grammar, alphabets, and loading them is presented as
   ```
   Dim ArabicGrammar As New
     Recognition.SrgsGrammar.SrgsDocument
   Dim AlphabetRule As New
     Recognition.SrgsGrammar.SrgsRule("ABTS")
   Dim AlphabetList As New
     Recognition.SrgsGrammar.SrgsOneOf ("Alif",
     "Baa", "Taa",, "Thaa", "Jeem", "Haa", "Khaa",
     "Daal", "Thaal", "Raa", "Zaa", "Seen", "Sheen",
     "Sahd", "Dahd", "Ttaa", "Thaa", "Ayn", "Ghayn",
     "Faa", "Qaaf", "Kaaf", "Laam", "Meem",
     "Nuun", "Ha", "Waaw", "Yaa", "Sifer", "Wahid",
     "Ithnain", "Thalatha", "Arbaa", "Khamsa",
     "Sitta", "Sabaa", "Thamaniya", "Tisaa")
   AlphabetRule.Add(AlphabetList)
   ArabicGrammar.Rules.Add(AlphabetRule)
   ArabicGrammar.Root = AlphabetRule
   ABTS.LoadGrammar(New
     Recognition.Grammar(ArabicGrammar))
   ```
4. The events, whether the defined *ABTS–ArSL* is recognized according to the grammar prescribed, or otherwise and these require a subprogram defined as
   ```
   Private Sub ABTS_SpeechRecognized(ByVal
     sender As Object, ByVal e As
     System.Speech.Recognition.RecognitionEventArg
     s) Handles ABTS.SpeechRecognized
   ```
   which returns the value in *e* as *e.Result.Text*

5. Each value in Step 4 is connected to the chosen location to display the sign and avatar for each alphabet designated in Step 3. For example, the case for "*Alif*", the images are displayed upon executing these statements

   pbxArabic.Image = System.Drawing.Image.FromFile ("D:\ABTS\Letters-Arabic
   pbxSign.Image = System.Drawing.Image.FromFile ("D:\ABTS\Letters-Sign\1-alif.jpg")
   pbxABTS.Image = System.Drawing.Image.FromFile ("D:\ABTS\Letters-Avatar\1-alif.jpg")

   The message "What was that?" is displayed when no matching is attained.

3.3 Testing the *ABTS–ArSL*

ABTS–ArSL is developed using *MS-VB*2008 which provides the test for capturing voice of a user pronouncing the 28 alphabets and numbers (0 to 9). The source codes are given in *MS-VB*2008 form environment as in Fig. 8, 9 and 10.

Fig. 11 shows *ABTS–ArSL* main menu upon executing the system. A user is prompted to utter the words "Start listening" as shown in one of the pop-up window (Fig. 6). The speech recognition could also be invoked by clicking the "On" command in its main menu (Fig.5).

For a word uttered with correct phoneme, a pop-up window shows the word and *ABTS–ArSL* displays the matching alphabet or number sign and its avatar image as shown in Fig. 12 and 13.

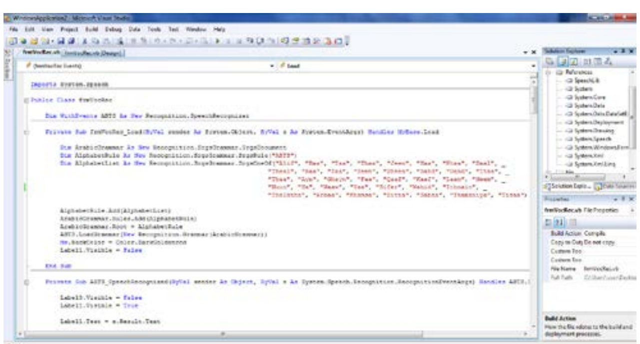


Fig. 8 The *ABTS–ArSL* grammar and definition in *MS-VB*2008 Form

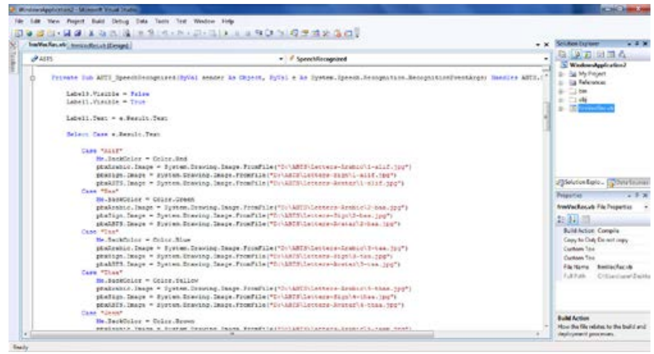


Fig. 9 The *ABTS–ArSL* recognizer and speech analysis

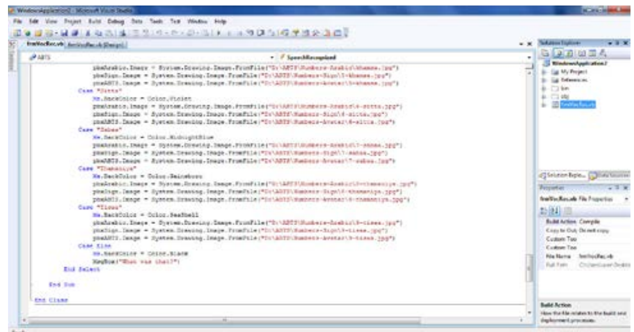


Fig. 10 End of *ABTS–ArSL* speech analysis

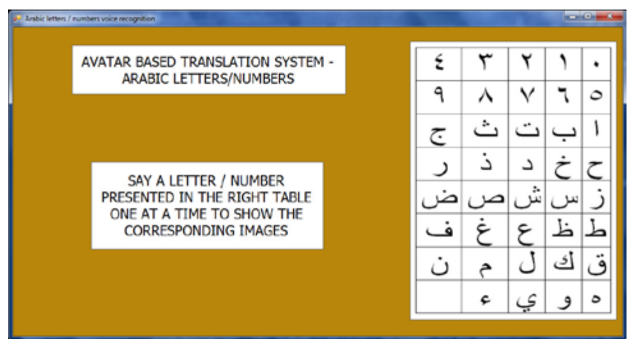


Fig. 11 *ABTS–ArSL* Main menu

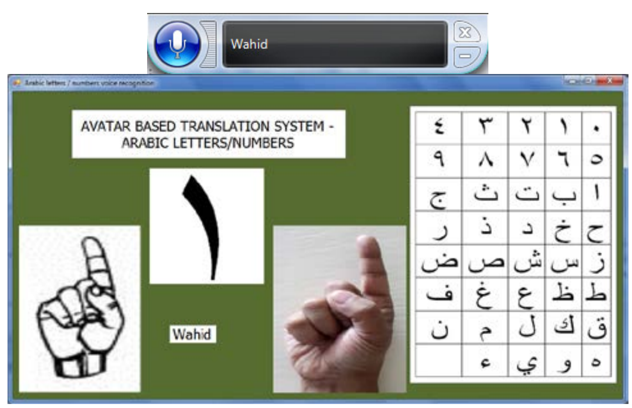
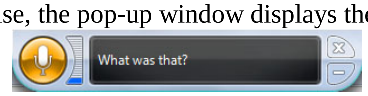


Fig. 12: The Sign, Avatar and pop-up message for Number 1

Fig. 13: The Sign, Avatar and pop-up message for Alphabet Sheen

Otherwise, the pop-up window displays the message:



with a "beep" sound.

Fig. 14, 15, 16 and 17 are the outputs shown by the correct pronunciation for number 7, alphabets "Jeem", "Ghayn" and "Qaaf". The respective pop-up windows are not shown as it has been displayed in Fig. 12 and 13.

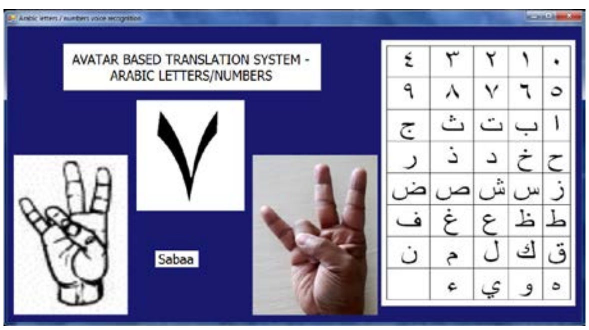


Fig. 14: The Sign and Avatar for Number 7

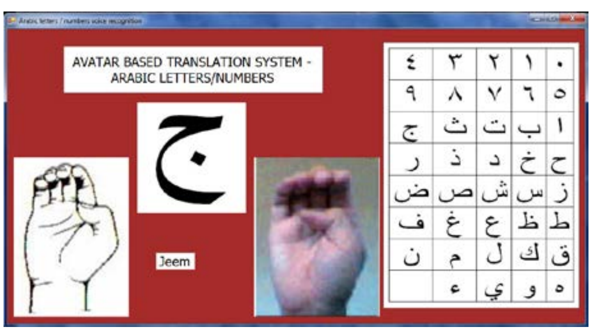


Fig. 15 The Sign and Avatar for Alphabet "*Jeem*"

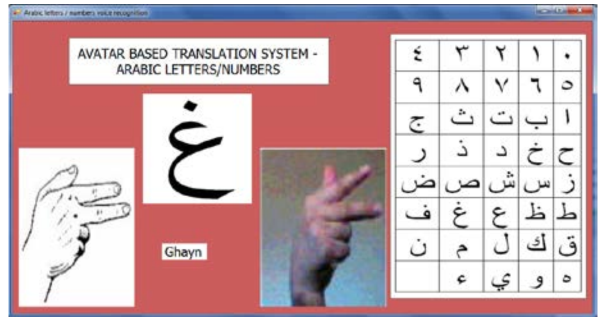


Fig. 16 The Sign and Avatar for Alphabet "*Ghayn*"

Fig. 17 The Sign and Avatar for Alphabet "*Qaaf*"

The 28 consonant phonemes Arabic alphabets and numbers (0 to 9) have phonemic contrasts between bilabial (both lips), emphatic (pharyngeal or velar), and glottal. For the non-emphatic alphabet likes "*Baa*" and "*Jeem*" which sound similar to English alphabet "*B*" and "*J*", require the use of both lips and the pronunciation are manageable. Some difficulties arose when uttering a word emphatically or glottal. The Number "*Wahid*", the Alphabets "*Ghayn*" and "*Qaaf*" require a precise 'makhraj' to utter these words. Fig. 18 shows the placement of Arabic alphabet in details.
A clear recognition to the alphabets that pronounced with bilabial (both lips) is contributed to the natural pronunciation of those alphabets. The harder recognition to the alphabets in pharyngeal, velar and glottal could be overcome by a proper training in order to attain a perfect phoneme for those alphabets. Fig. 18 shows the exact location for the alphabets "*Alif*", "*Haa*", "*Khaa*", "*Ayn*", "*Ghayn*" and "*Ha*" which have a harder pronunciation.

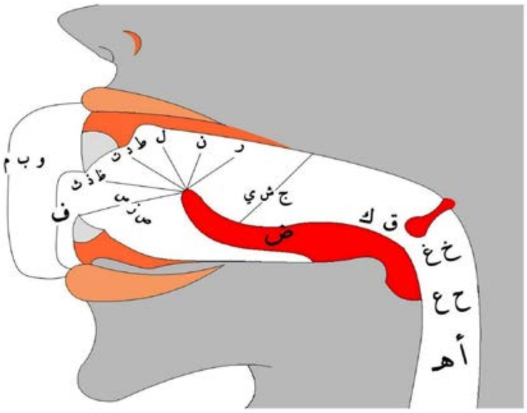


Fig. 18: The Arabic alphabets phoneme placement

4. Conclusions and Recommendations

Development of speech recognition is widely used in industrial software market but the display of the image of speech might be not readily available. Since the sign language recognition is still an active area of research this will surely continue to evolve. In this paper we have described the *ABTS–ArSL* system to translate a speech in Arabic directly to avatars. For this research, the biggest challenge is developing the speech recognition system. The accuracy of capturing Arabic speech and deciphering it correctly is vital for the success of the following processes. The building of the sign and Avatar databases are feasible with the right resources.

The proposed phonemes recognition in *ABTS–ArSL* encourages a user to perfectly utter the Arabic alphabets and numbers. Also, it can assist beginners to start their lessons on how to pronounce the alphabets and numbers correctly and then gradually spell out phonetically. The process becomes more effective after a frequent training. In this research, the built grammar confined only to a single word representing the Arabic alphabets/numbers and can be extended further. Windows speech recognition engine may be utilized to make room for a double or triple words combination and the grammar enhancement might be possible. The development of a new grammar in other languages with its related signs and avatars are also another possible future research.

Building an accurate automatic sign language recognition system is of a great importance in facilitating efficient communication between normal and deaf people. The *ABTS–ArSL* is a visual natural language model that might be used for communication by both normal and the deaf community in Arabic speaking countries.

Acknowledgments

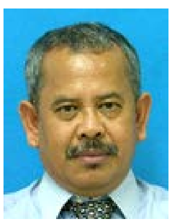
This project was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah under grant no 27–011/430. The authors, therefore, acknowledge with thanks DSR technical and financial support.

References

[1] http://www.worldometers.info/world-population/, 18/12/2013
[2] http://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers, 18/12/2013
[3] S. M. Halawani and Zaitun A.B., "An Avatar Based Translation System from Arabic Speech to Arabic Sign Language for Deaf People", International Journal of Information Science and Education. Vol 2, No. 1, pp. 13-20, 2012.
[4] http://en.wikipedia.org/wiki/American_Sign_Language, 18/12/2013
[5] A. Farhadi, D. Forsyth, and R. White. "Transfer Learning in Sign Language", IEEE Conference on Computer Vision and Pattern Recognition, CVPR '07, pp: 1 - 8. 2007.
[6] http://en.wikipedia.org/wiki/British_Sign_Language, 18/12/2013
[7] S. Cox, M. Lincoln, J. Tryggvason, M. Nakisa, M. Wells, M. Tutt, and S. Abbott, "Tessa, a system to aid communication with deaf people", Proceedings of the fifth International ACM Conference on Assistive Technologies, Assets '02, ACM New York, NY, USA, 2002.
[8] M. A. Abdel-Fattah, "Arabic Sign Language: A Perspective", Journal of Deaf Studies and Deaf Education, Vol. 10, No. 2. Oxford University Press. 2005.
[9] S. Halawani, "Arabic Sign Language Translation System on Mobile Devices", International Journal of Computer Science and Network Security, Vol.8, No.1, pp 251-256 2008.
[10] N. El-Bendary, H.M. Zawbaa, M.S. Daoud, A. ella Hassanien, K. Nakamatsu. "ArSLAT: Arabic Sign Language Alphabets Translator", International Journal of Computer Information Systems and Industrial Management Applications, Vol 3; pp. 498-506, 2011.
[11] S. F. Ghanem, "An Avatar Based Natural Arabic Sign Language Generation System for Deaf People", M.Sc. Thesis, King Abdulaziz University, Jeddah, Saudi Arabia, 2008.
[12] H. Satori, M. Harti & N. Chenfour. "Arabic Speech Recognition System Based on CMUSphinx", International Symposium on Computational Intelligence and Intelligent Informatics, ISCIII '07, pp. 31-35, 2007.
[13] H. Al-Dosri, N. Alawfi and Y. Alginahi, "Arabic Sign Language Easy Communicate ArSLEC", 1st Taibah University International Conference on Computing and Information Technology, ICCIT 2012, pp. 474-479, 2012.
[14] Khaled Assaleh, M. Al-Rousan. "Recognition of Arabic Sign Language Alphabet Using Polynomial Classifiers", EURASIP J. Adv. Sig. Proc, Vol 13, pp. 2136-2145, 2005.
[15] M. Mohandes, S. I. Quadri, M. Deriche, "Arabic Sign Language Recognition an Image–Based Approach", IEEE 21st International Conference on Advanced Information

Networking and Applications Workshops (AINAW'07), Vol 1, pp. 272 – 276, 2007.
[16] http://www.linguanaut.com/arabic_alphabet.htm, 18/12/2013
[17] http://en.wikibooks.org/wiki/Arabic/Arabic_sounds, 21/12/2013
[18] http://mylanguages.org/arabic_alphabet.php, 18/12/2013
[19] http://wikitravel.org/en/Arabic_phrasebook, 19/12/2013
[20] Microsoft Windows 7 Professional Operating Systems
[21] http://msdn.microsoft.com/en-us/library/, 12/12/2013
[22] http://www.islamicposters.co.uk/how-to.html, 12/12/2013
[23] http://social.msdn.microsoft.com/Profile/bruno yu/activity, 12/12/2013

Sami M. Halawani received his M.S. degree in Computer Science from the University of Miami, USA, in 1987. He received the Professional Applied Engineering Certificate from The George Washington University, USA, in 1992. He earned the Ph.D degree in Information Technology from the George Mason University, USA in 1996. He is an Associate Professor in Faculty of Computing and Information Technology Jeddah, King Abdulaziz University Jeddah, Saudi Arabia.



Daut Daman received his Master degree in Applied Mathematics from Cranfield University, England in 1984. He was an Associate Professor in Faculty of Computing, Universiti Teknologi Malaysia, Malaysia. In 2009, he joined Faculty of Computing and Information Technology Rabigh, King Abdulaziz University Rabigh, Saudi Arabia as an Associate Professor**.**

Sarudin Kari received his M.Sc. degree in Information Systems from Universiti Teknologi Malaysia, Malaysia in 1989. In 2009, he joined Faculty of Computing and Information Technology, Rabigh, King Abdulaziz University Rabigh, Saudi Arabia as an Associate Professor. At present, he is an Associate Professor in Faculty of Computing, Universiti Teknologi Malaysia, Malaysia.

Ab Rahman Ahmad received his M.Sc. degree in Optimization and Computing from Loughborough University of Technology, England in 1985. His Ph.D degree in Elliptic Partial Differential Equations is also received from Loughborough University of Technology in 1993. He was an Associate Professor in Faculty of Computing, Universiti Teknologi Malaysia, Malaysia. In 2009, he joined Faculty of Computing and Information Technology Rabigh, King Abdulaziz University Rabigh, Saudi Arabia as an Associate Professor**.**