Arabic Sign Language Translation System On Mobile Devices

Dr.Sami M.Halawani

Information Technology Dept., College of Computing and Information Technology,
King Abdulaziz University, Jeddah, Saudi Arabia

Summary
Sign Language Translation System/software that translates text into sign language animations could significantly improve deaf lives especially in communication and accessing information. In the last few years the usages of technology have increased rapidly. One of the most popular technologies is the invention of mobile devices. The most common application is mobile phones for voice transmission, but systems for data transmission are also available. The Wireless Application Protocol (WAP) is an open, global specification that empowers mobile users with wireless devices to easily access and interact with information and services instantly. Previous text-to-Sign Language projects have made limited progress by restricting their output to the PC base – thus avoiding important animation issues and mobilizations of people. This paper introduce Arabic Sign Language Translation Systems (ArSL-TS) Model that runs on mobile devices

Key words:
deaf, Arabic, sign language, translation, mobile devices, WAP, HCI

1. Introduction

Signing has always been part of human communications. The use of gestures or sign is not tied to ethnicity, age, or gender. Infants use gestures as a primary means of communication until their speech muscles are mature enough to articulate meaningful speech. For millennia, deaf people have created and used signs among themselves. These signs were the only form of communication available for many deaf people. Within the variety of cultures of deaf people all over the world, signing evolved to form complete and sophisticated languages. These languages have been learned and elaborated by succeeding generations of deaf children. Normally, there is no problem when two deaf persons communicate using their common sign language. The real difficulties arise when a deaf person wants to communicate with a non deaf person. Usually both will get frustrated in a very short time. For this reason, there have been several attempts to design smart devices that can work as interpreters between the deaf people and others. These devices are categorized as human-computer-interaction (HCI) systems [5].

In recent years, several research projects in developing sign language animations system have been developed [2]. Some previous projects have made efforts in translating English text into Sign Language Animation, but none have proposed practical systems for translating Arabic text into Arabic Sign Language use mobile technologies and mobile devices. And also, most of the previous systems are PC-base. The adoption of mobile devices in developing sign language animation systems is motivated by several considerations: they help deaf to upgrade quality of human-human communication by evolving animations; they have a positive impact on factors such as human-mobility and likeability; they can have a positive effect on a deaf perception of deaf learning experience because they can attract deaf attention.

The adaptation of mobile devices makes sign language translation more attractive and more valuable. Using mobile devices instead of PC-base in sign languages presents several advantages. With PC base, term of anywhere and any place in deaf learning cannot be applied. With mobile devices one can obtain communication more realistic, a wider usage of sign language applications becomes possible and practical.

During the last several years the usages of mobile services have increased rapidly. The most common applications are mobile phones and PDA for voice transmission and data transmission between mobile units or mobile units to server units, for example from mobile units to Internet is also available. The WAP is a global specification that empowers mobile users with wireless devices to easily access and interact with information and services instantly. WAP makes it possibly to communicate with the mobile user and to send and receive information [3]. This could be very useful for deaf in implementing of text translation into sign language animations through sign language animation systems. For example, deaf who want to translate certain

Manuscript received January 5, 2008
Manuscript revised January 20, 2008

word to correct sign language. The WAP programming model is similar to the World Wide Web (WWW) programming model. WAP defines a set of standard components that enables communication between mobile units and network servers.

The aim of this research is to present a model to develop an avatar based sign language translation system that allows users to translate Arabic text into Arabic Sign Language for the deaf on mobile devices such as PDAs. Figure 1 shows the general architecture in implementing of Arabic Sign Language Translation System (ArSL-TS).

2. Related Works

2.1 Sign Language Translation Systems

A recent development in the signing systems field is that of sign language animations, or signing avatars. Instead of a human person signing a story, a 3D animation (puppet and robot) is used to render the sign language on screen. This development may well have as large an impact on the 'signing systems' field. The objectives of 'sign language': a computer converts text into avatar or spoken text in real time, and without human intervention. In the case of 'sign language' systems however, the problem is more complex. It is not only a matter of converting printed text into another medium, but also into another *language*. The software therefore has to be a combination of translation software, and conversion software. To complicate matters even more, most sign languages have been described only partially. In many countries, sign language dictionaries are only now being developed, while research of the syntax and pragmatics of many national sign languages is just starting. Nevertheless, 3D translation systems are now being developed for two applications: sign language training programs and books in sign language in the USA, and for adding sign language to mainstream television broadcasts in the UK.
Two of such systems are briefly discussed next.

2.1.1 Seamless Solutions, USA

Seamless Solutions Incorporated develops the USA system. Several models have been developed (called 'avatars') including a signing frog. Examples in figure 2-1 to 2-3 show a dictionary application, a boy avatar signing a story, and a frog avatar. A recent project by Seamless Solutions (FACE, Facial Animation for Communication Enhancement) involved the inclusion of facial characteristics in an avatar in real time. An "Animation Engine" was developed to interactively control the facial expressions, in synchronisation with signing and optionally, voice.

The attraction of animated sign language is not (only) the fact that it enables frogs to sign. The animations are software generated, which allows for efficient storage and transmission, and flexibility in use. Once a sign specification is stored, it can be rendered on screen by different avatars, in all colours and sizes. The major driving force behind animated sign language, however, is computer-generated translation from written text or speech, into sign language.

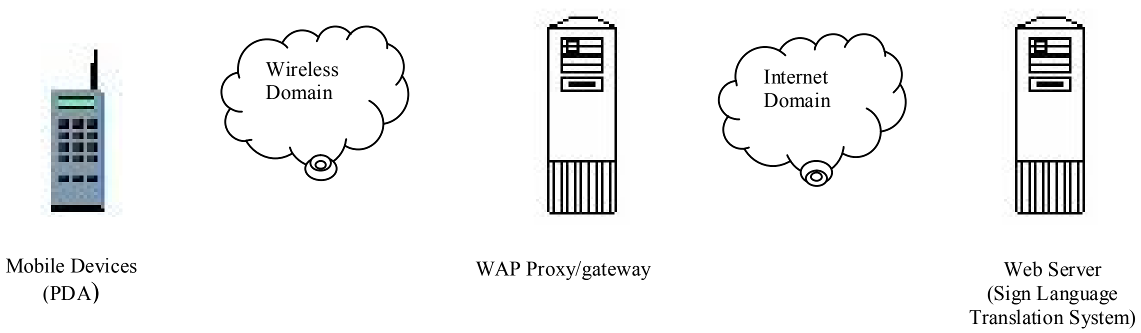

Figure 1: General Architecture of ArSL-TS

2-1 2-2 2-3

Figure 2-1 to 2-3: Examples of a boy and frog avatar

2.1.2 Simon, UK

In the UK, ITC (Independent Television Commission) has commissioned Televirtual to develop *Simon, the virtual signer* to translate printed text – television captions – into sign language, to solve some of the problems of adding sign language to television programs

The software consists of two basic modules: linguistic translation from printed English into sign language, and virtual human animation. The animation software allows Simon to sign in real-time. A dictionary of signed words enables the system to look up the accompanying physical movement, facial expressions and body positions, which are stored as motion-capture date (not images or video) on a hard-disk. The system can call these physical moves (in any order) and interpolate between them to create smooth, natural looking signing sequences. The motion-capture data – which includes hand, face and body information – is applied to a highly detailed 3D graphic model of a virtual human. This model includes very realistic and accurate hand representations, developed within the project. Natural skin textures are applied to the hands and face of the model to create the maximum impression of subjective reality.

2.2 PDA Sign Languages

A PDA Sign Language Editors is being developed as part of the VisInfo project which is the objective is to assist deaf people in their special needs [2]. With the development of this software, to be used in mobile devices, it is intended to avoid the obligation of learning oral language, in order to use a sign editing software. It is also intended, with the creation of this editor, specifically to mobile devices, to use this software as a communication tool between deaf people, or even, between deaf and non-deaf people, through infrared and Bluetooth data transmission devices, available in most of the PDA.

3. Sign Language on Mobile Devices

Sign languages, i.e. the visual communication languages used by deaf people, are an interesting application domain. For example, Karpouzis et al. [4] and Sagawa and Takeuchi [3] describe architectures and tools to teach sign languages, while Zhao et al. [6] propose an automatic machine translation system from written English to American Sign Language. The ViSiCAST project [8] aims at improving the accessibility of public services, e.g. by using monitors which display-embodied agents performing the sign language translation of what a Post Office counter clerk say. Yi et al. [10] proposed a tool to create and reuse animations for sign languages, using a specifically designed database.

3.1 Requirements for a Mobile Application Architecture

To provide the same capabilities as a non-mobile version like PC-base of the application, an application designed for mobility must support the following [7]:

i. *Consistent operation online or offline* – An application designed for mobility should enable users to access data, whether online or offline. When working offline, the user still perceives that the shared data is available for reading and writing. When network connectivity is restored, local data changes are integrated into the network copy of the data.

ii. *Seamless connectivity* – An application designed for mobility should work with an agent or proxy service to transparently handle changes in connectivity.

iii. *Multi-platform support for clients* – An application designed for mobility should, at a minimum, tailor its interaction and behavior to the characteristics of the current device, such as input or output type, resources available and performance level.

iv. *Optimized power use and performance* – An application designed for mobility should closely manage its use of the limited power of a portable device when battery operated.

3.2 The Sign Language

As in oral language, sign language is not universal; it varies according to the country, or even according to the regions. The year 1971 has been marked for deaf people in the world, when the Deaf Action Committee's Valerie Sutton presented Sign Writing. At the beginning, the graphical notation of Sign Writing had been created in order to write dance steps, but that was enough to earn attention from deaf language Danish researchers, that were looking for a system to write signs, today known as Sign Writing.

Sign language in the Arab World has recently been recognized and documented. Many efforts have been made to establish the sign language used in individual countries, including Jordan, Egypt, Libya, and the Gulf States, by trying to standardize the language and spread it among members of the Deaf community and those concerned. Such efforts produced many sign language**s**, almost as many as Arabic-speaking countries, yet with the same sign alphabets. Gestures used in Arabic Sign Language are depicted in Figure 3.

Figure 3: Example of Arabic Sign Language Alphabet (ArSL)

4. ArSL-TS Architecture Overview

ArSL-TS architecture comprises three types of functions:

- *Information sources* – Information sources provide data to the ArSL System, either on request using a "push" mechanism provided by "publish and subscribe message" middleware, or through a periodic extract, transform and load process. Information sources include databases of Arabic Sign, applications for translating text to sign animation, other external systems or library.
- *ArSL functions* – ArSL functions provide for aggregation and composition of the information and delivery to the user.
- *Independent functions* – Independent functions are pre-existing technologies or components, such as a Mobile Web browser or protocol proxy. They are generic functions, such as rendering of content in HTML/WML for display on a Mobile Web browser or traversing of networks and firewalls using a protocol proxy.

Components of a mobile ArSL-TS are:

- *Web browser* – Provides a user interface to the ArSL-TS. If access is over the Internet, a protocol proxy supports communication between the user and the portal using Hypertext Transfer Protocol (HTTP) and HTML/WML, often enhanced through the use of client-side scripting and/or browser-hosted code, such as ActiveX controls or Java.
- *Presentation/Application server* – Creates an integrated view of content through interactions with other components.
- *Sign language management, searching and indexing, and collaboration* – Manage the lifecycle of the sign language contents including modifications, such as creation, versioning and revisions.
- *Security* – A requirement for all architectures for mobile applications to ensure the integrity of sensitive information in remote locations. Today the protocols and interfaces between these components are proprietary to a product, but standards organizations and vendors are actively working to develop a set of XML-based standards.

To simplify the modeling of ArSL-TS and display them on mobile devices, a system composed of four main parts (Figure 4): (i) Mobile device player allows the user to input the text and to view the animation, (ii) the Text translator tool supports the translation of text in each phase communication process, (iii) Matching tool establish comparison and matching of text and sign, and (iv) the sign language stored in the Sign Animation Database.

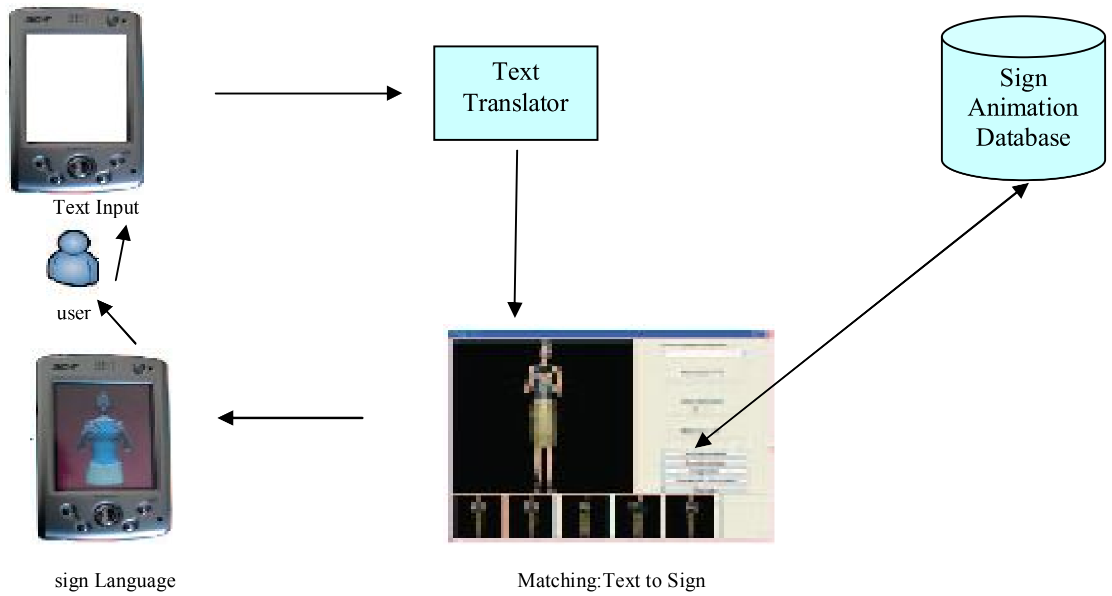

Figure 4: High-level diagram of ArSL-TS on Mobile Device

5. Conclusions

In this paper, we proposed the ArSL-TS for the text translating into sign language animations on mobile devices. Since ArSL-TS is intended for mobile deaf users (Arab people), we based it on a standard Arabic sign language and provide animation and instant feedback about the meaning of the Arabic text. It should be noted that there is a lot of room for further performance improvement considering different techniques and approaches.

References

[1] Bruno M.Vargas, Michael da S. Antunes, Paulo R. G. Luzzardi, Antônio C. da R. Costa , Glaucius D. Duarte, Ricardo A. Cava. PDA Sign Language Editors in Computer and Advanced Technology in Education 2006 –CATE 2006: IASTED Conference in Lima, Peru.

[2] Matt Huenerfauth. Generating American Sign Language Classifier Predicates For English-To Asl Machine Translation. Ph.D dissertation, University of Pennsylvania, Department of Computer and Information Science, Philadelphia, PA, USA 2006

[3] H. Sagawa and M. Takeuchi. A Teaching System of Japanese Sign Language Using Sign Language Recognition and Generation. In MULTIMEDIA '02: Proceedings of the 10th ACM international conference on Multimedia, pages 137–145. ACM Press, New York, NY, USA, 2002

[4] K. Karpouzis, G. Caridakis, S.E. Fotinea, and E. Efthimiou. Educational Resources and Implementation of a Greek Sign Language Synthesis Architecture. Computers & Education, Special Issue in Web3D Technologies in Learning, Education and Training, In press.

[5] Khaled Assaleh, M. Al-Rousan. Recognition of Arabic Sign Language Alphabet Using Polynomial Classifiers. EURASIP Journal on Applied Signal Processing 2005:13, 2136–2145

[6] L. Zhao, K. Kipper, W. Schuler, C. Vogler, N. I. Badler, and M. Palmer. A Machine Translation System from English to American Sign Language. In AMTA '00: Proceedings of the 4th Conference of the Association for Machine Translation in the Americas on Envisioning Machine Translation in the Information Future, pages 54–67. Springer-Verlag, Berlin, Germany, 2000.

[7] Luca Chittaro, Fabio Buttussi, Daniele Nadalutti. MAge-AniM. A system for visual modeling of embodied agent animations and their replay on mobile devices. 2005

[8] R. Elliott, J. R. W. Glauert, J. R. Kennaway, and I. Marshall. The Development of Language Processing Support for the ViSiCAST Project. In Assets '00: Proceedings of the 4th international ACM conference on Assistive technologies, pages 101–108. ACM Press, New York, NY, USA, 2000.

[9] Thomas Lindberg, Johan Uneman. WAP Research. Master Thesis, School of Electrical and Computer Engineering, Chalmers University of Technology, October 2001
[10] Yi, J. Frederick C. Harris, and S. M. Dascalu. From Creating Virtual Gestures to "Writing" in Sign Languages. In CHI '05: Extended abstracts on Human factors in computing systems, pages 1885–1888. ACM Press, New York, NY, USA, 2005.

Dr. Sami M. Halawani Received the 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 a faculty member of the College of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia. He is currently working as the Vice Dean for Graduate Studies and Research.