MULTIMODAL INTERFACES TO THE COMPUTER

Hema A. Murthy and C. Chandra Sekhar
Department of Computer Science and Engineering, IIT Madras Chennai , India

C.S. Ramalingam
Department of Electrical Engineering, IIT Madras Chennai , India

Srinivasa Chakravarthy
Department of Biotechnology, IIT Madras Chennai , India

Keywords: Multimodal interface, Indian languages, speech recognition, handwriting recognition, keyboard interfaces, loadable kernel modules.

Contents

Summary

The literacy levels in Asia and Africa range from 50-30%. In particular, in Africa the literacy level is as low as 50% in most nations. Even though the world-wide web and computer communication has given us access to information at the click of a mouse, 95% of the world's population is excluded from this revolution due to dominance of English. A multimodal interface to the computer that is relevant for developing nations overcomes this problem. The components of this interface are: (a) Keyboard and display interface, (b) Speech interface, and (c) Handwriting interface. The chapter first motivates the need for multimodal interfaces to the Desktop computer with India as an

example. The issues in building each of these interfaces are then addressed. Finally, a case for building these interfaces into the operating system is presented, so that they become available for all applications, obviating the need to recompile applications when a language change is needed.

1. Introduction

Imagine a villager walking into a rural Internet kiosk, who may be semi-literate or even literate, wanting to use the power of the Internet to either communicate with a relative somewhere else, or contact a city hospital, or get vital crop information. The currently available English-based keyboard and applications are totally unfamiliar and intimidating. As a result he or she feels shut out and is not a part of the ongoing information revolution. On the other hand, if the computer had applications in the local language, computers would become part of the daily lives of the average villager.

However, making local language interfaces to the computer is not an easy task. For example, in a typical Indian language, there are roughly 3000 character clusters. The number of keys on the keyboard will become unmanageably large if we want a single keystroke for each cluster. Instead, we will have to make do with a sequence of keystrokes, making even typing a difficult task. Even a literate person will find this difficult. The keyboard is, therefore, not an easy-to-use interface for many languages.

It is essential to have an interface that not only uses the local language but also is more natural to use. We use speech and handwriting for communicating to others, and hence these are the most natural interfaces for a computer to have. The handwriting interface, while making data entry in Indian languages much easier than typing using the QWERTY keyboard; it can only be used by a literate person. On the other hand, the speech interface makes the computer and the Internet accessible even to the semi-literate and illiterate sections of the population.

An interface using keyboard, handwriting, and speech is called a multimodal interface. It is important to realize that the speech and handwriting interfaces supplement the keyboard, not replace it.

Although speech and handwriting interfaces are available for English [1], these are for some specific applications only, e.g., dictation machines, limited handwriting/graffiti recognition (in Personal Digital Assistants or PDAs), etc. This is primarily because of the simplicity of the Roman script. Whereas, for many other languages, these interfaces must be part of any application, viz., mail readers, web browsers, word processors, etc. Further, the interfaces must work seamlessly with any application. To facilitate this, modifications are needed at the operating system level, so that any application that runs on top of the OS can inherit the interface.

2. Issues in providing Multimodal Local Language support to the Computer

In this section we highlight some of the main issues involved in providing keyboard, speech, and handwriting interfaces to the computer. Examples from Indian languages are given to illustrate some of the issues.

2.1. The Keyboard Interface

The current keyboard is designed for English. The efforts for developing keyboards for Indian languages have been exclusively remapping the keys of the QWERTY keyboard. Although the key-board is more reliable and robust than the speech and handwriting interfaces, it is cumbersome to type in languages whose orthographic representations are grapheme 1 rich. For example, in Indian languages, there are 3000 different graphemes, requiring multiple key strokes in the remapped QWERTY keyboard.

Instead, the keyboard must be designed from scratch, keeping in mind the similarities that exist between all the Indian languages. If there are important differences, those must also be taken into account and localized accordingly. For example, Tamil does not have aspirated consonants, which frees up keys that can be used for some of the most frequently occurring character clusters. A key idea in the design of the multimodal interface is separating the language-dependent and language-independent parts of any application. The language-dependent aspects are made part of the language resource. As an example, the static strings in the drop-down menu of an application are stored in the required language in an appropriate format and brought in when a change to that language is sought. This is similar to what is done for European languages. It is important to realize that the keyboard interface will continue to play a key role and not be replaced by the speech and handwriting interfaces. This is because the recognition accuracy for speech and handwriting is not 100%. Therefore, the keyboard must be available as a fallback option. Even if the recognition accuracy improves because of advances in technology, there are many instances when making one or more keystrokes is easier and quicker than achieving the same task by speech or handwriting. Hence the goal is to supplement the keyboard with the speech and handwriting interfaces, and not replace it.

2.2. Speech Recognition

The goal of the speech interface is to recognize speech for both data input as well as command and control. Typical data entry tasks are typing up a document in an Indian language. Therefore, applications like Dragon Naturally Speaking" [1] or Via Voice" [2] need to be developed for Indian languages. Typical command and control tasks are change directory, open a file, etc. Some of these may be more easily carried out using a few quick keystrokes. Hence the speech interface will coexist with the keyboard and not replace it. Clearly continuous speech recognition is the core technology for the input interface.

To be able to make progress in speech recognition having standardized databases is an essential pre-requisite. One of the major issues in Speech Recognition for Indian languages is the lack of such databases. In the US, a variety of databases for English have been collected over two decades, in a variety of conditions (clean speech, telephone, cellular, etc.) [3], spanning different applications. By contrast, we do not have a collection that is even remotely close for even one Indian language. Collecting such databases is a significant effort and we need a concerted effort to rectify this basic deficiency. Databases are also required for speech synthesis, but the requirements are different. Before collecting a database, it has to be carefully designed by an expert

linguist, taking the application into consideration. Annotation of the collected data requires careful development of a number of tools (see Section 4).

2.3. Handwriting Recognition

Although local language computing will enable the non-English speaking public to use the computer, the corresponding keyboard is very cumbersome to use. As we pointed out earlier, a typical Indian language has roughly 3000 character clusters (here by character cluster we mean the C*V or V (C stands for “consonant" and V for “vowel"). On a keyboard that is restricted to 256 possible scancodes, it is not possible to avoid multiple keystrokes for most of the clusters, except perhaps the most frequently occurring ones. An attractive alternative is the handwriting interface.
Handwritten character recognition can be classified as (i) scanned image of handwritten text on paper, and (ii) handwritten text produced by an electronic pen, where the pen trajectory on a special tablet is processed by the computer. The latter is known as Online Handwritten Character Recognition (OHCR). A handwritten character is composed of a set of pen strokes. A stroke is a line drawn by a pen between the time when a pen touches the writing surface and the time when it is lifted. Therefore, identifying the component strokes of a character is the first step to character recognition. The following issues need to be addressed:

* Database of Strokes: Stroke databases for all Indian scripts are not available. In stroke databases each grapheme is represented by a sequence of strokes [4]. In order that the OHCR is robust, graphemes need to be collected from a number of different writers. As each grapheme is made of a sequence of strokes, it is sufficient to collect the strokes that are possible in all languages. The stroke database can be made language independent. The sequence of strokes that take up the graphemes in a language can be represented by a grammar. The language generated by the grammar corresponds to the graphemes.

* Interfaces: An OHCR system consists of four main components. The (i) input area where the contents are entered, (ii) the stream of stroke IDs, (iii) the stream of characters, and (iv) the output. Between every stage and its successive one, interfaces must be defined and standardized. Standard interfaces are the key to the longevity of associated software.

3. Output Mechanisms

In this section we discuss different output mechanisms. Although humans have five different sensors the focus here is on the visual and auditory sensors.

3.1. Display

Unlike in English, the display corresponding to a keystroke is not only dependent upon the past keystrokes but also on the future. Enabling such support requires a language model for the representation of graphemes. For this, the kernel must be enhanced to support such language models. Additionally, the widths of the graphemes vary significantly, making variable-width fonts essential (fixed-width fonts will give ugly

results). There are a number of issues associated with variable-width fonts that we propose to address.

As mentioned earlier, a sequence of keystrokes may have to be pressed to generate a single character cluster. Further, as one types, the cluster will have to be modified to ensure that it corresponds to the key sequence in progress. This brings in the related issue of the positioning of the cursor. Today rendering engines [5] are available for proper rendering in any major Indian language provided the encoding is in UNICODE [6]. But there is a huge proliferation of a number of different fonts for each language. Each font requires that the font-based encoding is converted to UNICODE for rendering purposes.

Currently the cursor positioning is handled by the application. Since most applications are developed for English/European languages, applications will have to be modified to support Indian languages. This would result in every application being modified. To avoid this, the keyboard driver can be modified. Depending upon the language that is active, the rendering engine [5] ensures that the appropriate characters are displayed.

Once the interface is available, applications need to be customized for each language (menu bars, help, etc.). Sometimes applications need to be modified to separate the language dependent parts from the language independent parts.

-
-
-

Bibliography

[1] http://www.nuance.com/naturallyspeaking/ [Commercial speech recognition software]

[2] http://www.nuance.com/viavoice/ {commercial speech recognition software]

[3] http://www.ldc.upenn.edu [NIST database]

[4] Aparna K H, Vidhya Subramanian, Kasirajan M, Vijay Prakash G.V.S. Chakravarthy, Sriganesh Madhavanath, " Online Handwriting Recognition for Tamil," in Proceedings of the IWFHR-9 2004. [Online handwriting recognizer]

[5] http://gtk2-perl.sourceforge.net/doc/pod/Gtk2/Pango/Renderer.html [Renderer for local languages]

[6] http://www.unicode.org

[7] http://www.cstr.ed.ac.uk/projects/festival, Festival Speech Synthesis System [Speech synthesis software]

[8] N Sridhar Krishna, "Multilingual Text-to Speech Synthesis," MS Thesis, Dept. of Computer Science and Engineering, IIT Madras, 2004. [Diphone based speech synthesizer for Indian languages]

[9] M. Nageshwara Rao, S. Thomas, T. Nagarajan and H. Murthy, "Text-to-speech synthesis using syllable-like units," NCC-2005, IIT Kharagpur, India, Jan 2005, pp 277-280. [A syllable-based speech synthesizer for Indian languages]

[10] Neti, C.,Iyengar, G.,Potamianos,G., & Senior,A. (2000). Perceptual interfaces for information interaction: Joint processing of audio and visual information for human-computer interaction. In B. Yunan, T.Huang & X.Tang (Eds:), Proceedings of the International Conference on Spoken Language Processing(ICSLP 2000),Vol.3, (pp 11-14). [Audio visual multimodal interfaces]

[11] Neal, J.G.,& Shapiro, S.C.(1991). Intelligent Multimedia interface technology. In J.Sullivan & S.Tyler (Eds:), Intelligent User Interfaces (pp:11-14).New York: ACM Press. [Multimodal interfaces]

[12] Seneff, S.Goddeau, D.Pao,C., & Polifroni, J. (1996). Multimodal discourse modelling in a multi-user multi-domain environment. In T.Bunnell & W.Idsardi(Eds.), Processing of the International Conference on Spoken Language Processing,Vol.1 (pp:192-195). University of Delaware& A.I.duPont Institute. [Multimodal discourse modeling]

[13] Minh Tue Vo and Cindy Wood, Building an Application Framework for Speech and Pen Input Integration in Multimodal Learning Interfaces, Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 1996 (Atlanta; GA;May 1996). [Speech and pen interfaces]

[14] J Patricia, K Ratheesh, V S Shenoi, G Sreepriya, T A Gonsalves and Hema A Murthy, "Indian Language Support for the Linux Operating System," Int'l Symposium on Information Technology, People's Development and Culture, Allahabad, February, 12-16, 2001. [Keyboard based Indian language support]

[15] Anitha Nalluri, Bala Saraswathi A, Bharathi S, Hema A Murthy, Patricia J, Timothy A Gonsalves, Vidhya M S, Vivekanathan K, " Indian Language Support for X - Window System," in Proceedings of the ICON 2002, (SP-P6.4, May 2004). [Keyboard based Indian language support]

Biographical Sketches

Dr. Hema A. Murthy received her B.E. (1980) in Electronics and Communications Engineering from Osmania University, Hyderabad, India, an M.Eng (1986) in Electrical and Computer Engineering from McMaster University, Canada, and a Ph.D (1992) in Computer Science and Engineering from Indian Institute of Technology Madras (IIT-M), India. From 1980 through '83, she was a Scientific Officer with the Speech and Digital Systems Group of the Tata Institute of Fundamental Research, Bombay, India. In 1988, she joined the faculty of the Department of Computer Science and Engineering, IIT-M, India. Her current interests include speech recognition and synthesis, computer networks, multimodal interfaces to the computer.

C. Chandra Sekhar received the B.Tech. degree in Electronics and Communication Engineering from Sri Venkateswara University, Tirupati, India in 1984. He received the M.Tech. degree in Electrical Engineering and the Ph.D. degree in Computer Science and Engineering from Indian Institute of Technology Madras, Chennai, India, in 1986 and 1997, respectively. Since 1989, he has been a member of the faculty in Department of Computer Science and Engineering, Indian Institute of Technology Madras, and currently he is working as an Associate Professor. From June 2000 to May 2002, he was a JSPS Postdoctoral Fellow at Itakura Laboratory, Department of Information Electronics, Nagoya University, Nagoya, Japan. His current research interests include speech recognition, artificial neural networks and support vector machines.

C.S. Ramalingam obtained his BE from Madras University (1985), M.Tech from IIT Kharagpur (1987), and the Ph.D degree in Electrical Engineering from the University of Rhode Island (1995). In 1988 he was an Associate Lecturer at VLB Janaki Ammal College of Engineering and later in Kumaraguru College of Technology, both in Coimbatore. From 1995 to 2001 he was with the DSPS R&D Centre at Texas Instruments, Dallas, working in the areas of Speech Recognition and Speech Coding. He is currently an Assistant Professor in the Dept. of Electrical Engineering at IITM. His areas of interest are Signal Processing, Speech Recognition, and Speech Coding.

V. Srinivasa Chakravarthy received the B.Tech degree in Electrical Engineering from the Indian Institute of Technology, Madras in1989, and M.S. and PhD degrees from the Department of Electrical Engineering from the University of Texas at Austin in 1991 and 1996 respectively. He was a postdoctoral fellow until 1997 in the Neuroscience Division of Baylor College of Medicine, Houston, USA. He is

currently an Associate Professor in the Biotechnology Department, Indian Institute of Technology, Madras, India. Dr. Srinivasa Chakravarthy's current research interests include computational neuroscience and computational biology. Specific focus is on neuromotor modeling of handwriting generation, highlighting the role of basal ganglia, as a route to understanding Parkinson's disease. He is also involved in developing algorithms for handwritten character recognition for Indian languages.