FLEXIBLE SPEECH AND PEN INTERACTION WITH HANDHELD DEVICES

Sorin DUŞAN, Sriram RAMACHANDRAN

Center for Advanced Information Processing
Rutgers University
Piscataway, NJ 08854, U.S.A.
sdusan@caip.rutgers.edu, sriram@caip.rutgers.edu

Corresponding author: Sorin Duşan

An emerging research direction in the field of pervasive computing is to voice-enable applications on handheld computers. Map-based applications can benefit the most from multimodal interfaces based on speech and pen input and graphics and speech output. However, implementing automatic speech recognition and speech synthesis on handheld computers is constrained by the relatively low computation power and the memory limitations of these devices. This paper describes new results of our previous research on multimodal interaction using speech and pen inputs on handheld computers. Preliminary results show that combining speech input and pen gestures on handheld computers offers increased flexibility and improves naturalness and user satisfaction.

Key words: Speech interface, pen interaction, automatic speech recognition, gesture recognition.

1. INTRODUCTION

Mobile computers play an important role in the field of pervasive computing. From this category of computers, handheld devices offer the highest degree of mobility, but their computation (both speed and memory) and user interfaces are limited due to their size and power constraints. Handheld computers usually come with a pen operating on a touch screen and a soft input panel (SIP) displayed on the screen. An increasing number of other handheld devices, such as MP3 players, cell phones, play stations, etc., incorporate computing features and come with touch sensitive displays. Most of these devices have audio input/output capabilities; however often the quality of the microphones and loudspeakers is limited by their size and power. A natural enhancement to the user interface would be to add speech input/output modalities on top of the built-in audio capabilities. This has already passed from laboratory prototypes and demonstrations to everyday products. However, many such applications force users to interact by voice and pen in a specific (restricted) way.

Human-computer interfaces based on multimodal interaction have been developed for more than two decades, since the early system published by Bold, [1]. Some recent approaches are described in [2-5]. A review of the methods and systems for integrating speech and pen inputs is described in [6]. A comprehensive study of designing multimodal interactive maps for increasing human performance has been described in [7]. In our previous work we implemented a multimodal fusion method for desktop computer applications, primarily intended for fusing speech and mouse inputs as complementary streams of information, [8].

Systems of multimodal interaction for mobile computers usually employ speech input/output modalities. Integrating automatic speech recognition (ASR) and text-to-speech (TTS) synthesis on these small devices represents a challenge due to the memory and computation limitation of mobile computers. One solution is to implement these systems as client-server architectures using a wireless interface, and thereby to perform the high-computation ASR and TTS functions on the server side. Such approaches have been described in [9-11]. A multimodal system combining a personal digital assistant (PDA) for pen input and graphics output and a cellular phone for speech input/output is described in [12]. Another solution to speech-enabling mobile computers is to use an embedded speech engine. Such an approach is described in [13] using an ASR engine capable of recognizing a vocabulary of up to 500 active words. Previously we

presented an initial approach of an embedded solution for integrating simple speech and pen inputs on PDAs [14]. A criticism of this approach was that it only allows users to use simple pen taps in combination with speech. In this paper we present improvements to this initial approach.

2. MULTIMODAL INTERACTION BASED ON SPEECH AND PEN

PDA devices come in various sizes and with several operating systems. Among them, popular ones are the handheld computers based on the Pocket PC and the Palm OS platforms. These small devices have the main functionalities of a personal computer (PC) except for a few limitations due to their size. They do not usually have a keyboard, computer mouse or hard drive, and the memory and the microprocessor speed are also reduced, compared to desktop PCs. Although, special keyboards and hard drives can be attached to these computers, the computer mouse is replaced by a stylus (pen) operating on the device's touch screen. Such devices can run specialized applications for hand-character recognition using the pen as input. Most of these handheld PDAs also have built-in audio capabilities which enable the implementation of speech input/output modalities of interaction. Thus, the principal input modalities allowed by handheld computers are: pen on touch screen, buttons (including a navigation button) and audio (including speech). The possible output modalities are limited to graphics (including text, static graphics, and video) on the screen and audio (including speech).

Multimodal interfaces, including those on handheld computers, should offer the user the possibility of using various input modalities not only separately (independently) but also simultaneously (complementarily). In this section we show how to effectively exploit complementary speech and pen input modalities. In such constructions, each individual modality may or may not contain sufficient information for specifying a complete command. Speech and pen input modalities can be employed by users uni-modally (individually) or multi-modally (in combination with others). These options provide users with more natural and easy-of-use means of interaction.

2.1. Pen Taps and Speech

Due to the small size of the screen on PDAs text input is difficult and the speed for character entry is much less than when using a regular keyboard on a regular computer. These devices also do not have a permanent cursor on the screen, but the main functionalities of the computer mouse (pointing and selection) are accomplished using the pen input. A small number of buttons (present on most handhelds) also allow users to input, navigate and select various features. The built-in audio hardware, including a microphone and a tiny loudspeaker, allows audio recording and playback and enable ASR and TTS.

Fusing multimodal inputs on handheld computers requires special methods since these devices do not have a computer mouse and the pen does not entirely replace the full mouse functionality. The main difference consists in the lack of a cursor on the screen. The Windows CE or Windows Mobile operating systems treat the pen tap on the touch screen as a WM_LBUTTONDOWN Windows message (equivalent to a mouse click) and the WM_MOUSEMOVE message occurs only when moving the pen on the screen. Experimentally, we have found that for handheld devices it is easier to tap the screen at two different locations in order, for example, to refer to "this" and "here", than to move the pen between those two locations by keeping contact with the touch screen. For this reason we implemented the speech and pen tap fusion method using only WM_LBUTTONDOWN messages.

Another distinction from our previous fusion method for regular computers [8], is motivated by recent user studies that have shown that usually the gestures overlap with deictic utterances, but not all the time and not consistently across all users [2], [15]. Although most of the time gestures begin before speaking, this is not true all of the time and for all users. Thus, for some users speech precedes gestures by as much as 1-2 seconds [15]. These observations motivated us to implement a flexible technique for fusing speech and pen taps on handheld computers.

This new fusion method, originally introduced in [14] as an initial implementation, was designed especially for handheld devices but not exclusively (it can be adapted to run on PCs). It allows users to employ pen taps before, during or after speech utterances. Figure 1 shows a schematic time diagram depicting the three possible way of synchronizing pen taps with speech: (a) before speech, (b) during speech, and (c) after speech. The large arrows show the time positions of two pen taps that correspond to the deictic words "this" and "here" in a multimodal command involving speech ("Move this here") and two pen taps.

The multimodal command starts with the pen tapping on the microphone button at time t1. This initial pen tap (Pen Tap 0) switches the microphone to the ON position. The microphone remains in this state until an utterance is issued or until the pen again taps the microphone button thereby switching microphone into the OFF position. In all three cases the speech utterance "Move this here" starts at time t2 and ends at t3. After a short delay d1 from t3 the ASR engine turns off the microphone input and ends the input utterance at t4.

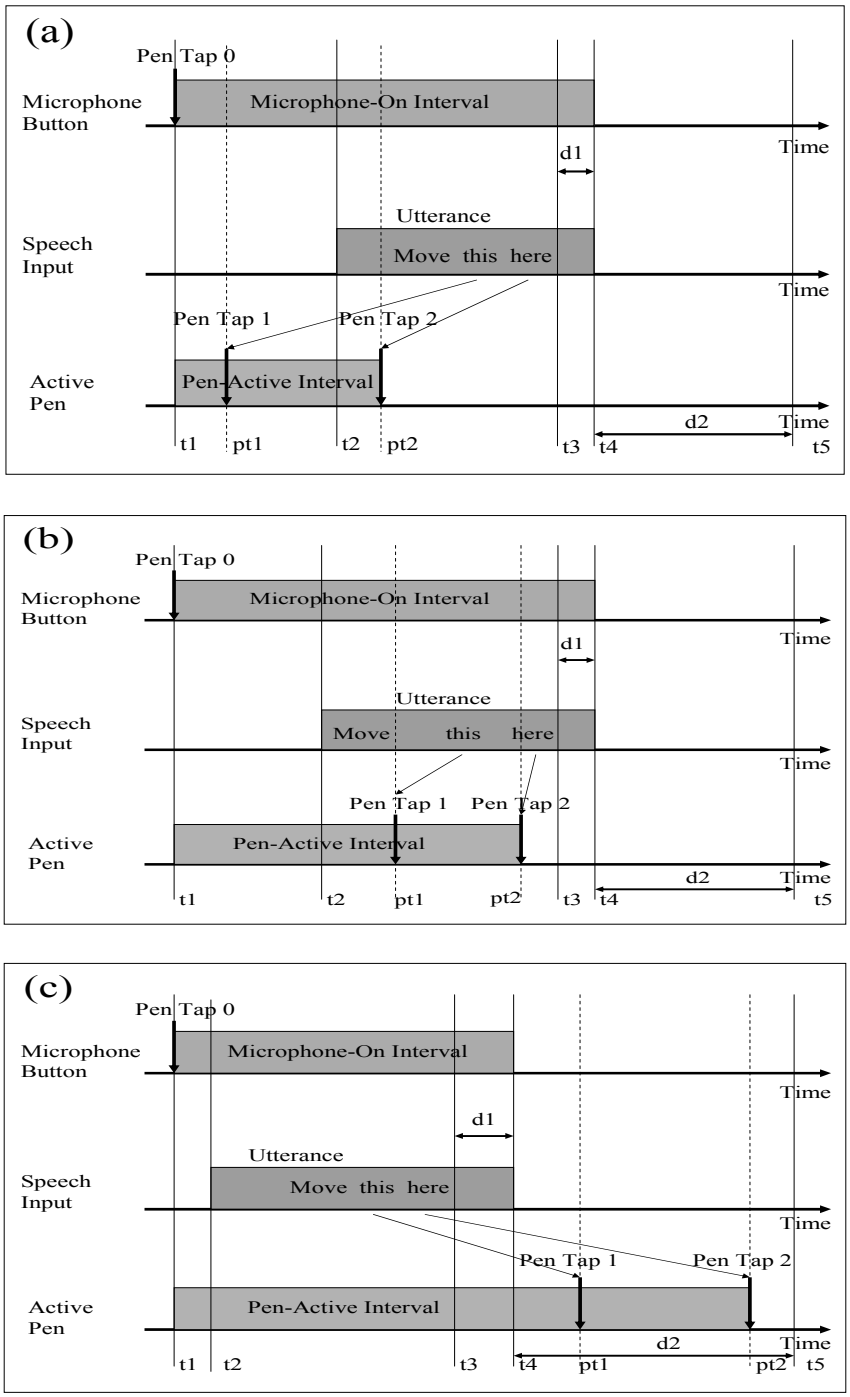

Figure 1. A time diagram showing the flexibility in synchronizing and fusing speech and pen inputs on handheld computers.

If the two pen taps (pt1 and pt2) needed for this spoken utterance do not arrive before t4 the system waits for a period of time d2 for these taps to occur (before t5). If both of them do not occur before t5 the spoken command is discharged. If more taps are received instead of two, only the last two are processed. In order to implement the flexibility in synchronizing speech and pen taps we built a buffer to store the pen tap positions for up to two deictic references in an utterance. However, in principle, this buffer can be increased to accommodate any number of pen taps during a single dialogue act. This buffer is emptied every time the user turns on the microphone by tapping the microphone button. This fusion method provides users with more flexibility in synchronizing multimodal constructions than in the previous implementations.

2.2. Speech and/or Pen Gestures

To respond to the previous criticism regarding the simplicity of the pen taps employed in our first implementation [14], we integrated a number of pen gestures in the current implementation. These pen gestures consist of single or double strokes and can be used in combination with speech input or alone.

In this implementation the single stroke gestures represent pen movements on the screen to specify directions. These are developed for obtaining the orientation (or final direction after rotation) of objects during object creation and object orientation (rotation). The four primary directions of left, right, upward and downward were implemented. For every pen stroke on the device, gesture values were calculated and buffered into a gesture buffer and a regression line is computed. The resulting direction is accepted within +/-30 degrees from the intended direction (0 degrees for right, 90 degrees for upward, 180 degrees for left, and 270 degrees for downward). If an utterance is recognized, the combination of the utterance and the gesture is referred against a lookup table. This table translates both speech and gesture inputs to a unique multimodal routine. Each pen gesture will be associated with a gesture value as follows: -1 (invalid), 0 (right), 1 (upward), 2 (left), 3 (downward), and 4 (double stroke). Once the unique routine is executed the gesture buffer is reassigned its invalid default value (-1) till another pen stroke occurs. If the speech recognizer first captures an utterance that can be combined with a gesture, the buffer value remains invalid till a gesture fills it. If the gesture buffer value does not translate to a valid gesture value within a configurable wait-time of 4 seconds, the object is created with a default direction (e.g. upward).

Figure 2 (left) shows an example of the screen when the user employed a speech ("Create a green car") and gesture (single stroke) command to create a green car picture oriented towards right. The green car object was created using a left-to-right pen gesture (single stroke) as indicated with an arrow mark.

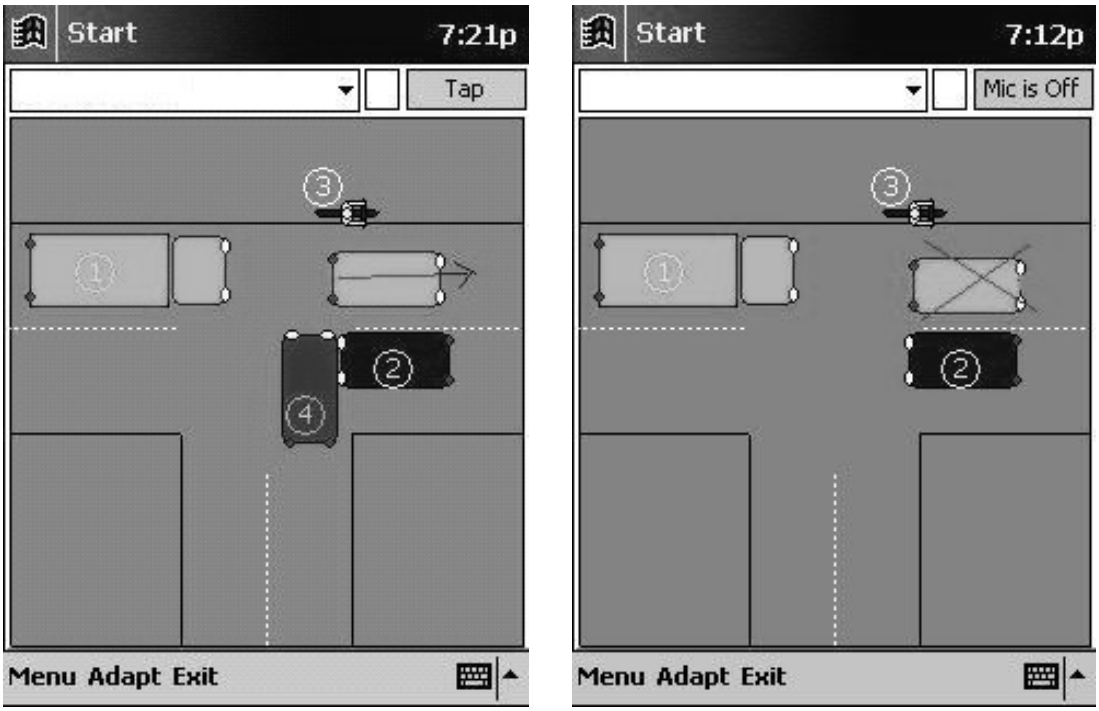

Figure 2. Example of creating (left) and deleting (right) a green car picture on the screen using speech and pen gestures.

The car is created at the centroid of the buffered points in our implementation but this can be changed to the starting point of the gesture as per user convenience. The same paradigm was applied to another directional command employed to change the orientation of objects. The specified object nearest to the centroid of the buffered points is chosen as the object to be acted upon.

A double stroke gesture was defined in this application to delete objects from the screen. In order to recognize a gesture with more than one stroke, it was not enough to have just spatial information on the screen. Windows CE and Windows Mobile provide a function that retrieves the number of milliseconds that have elapsed since the system was started. This function is used to distinguish multiple strokes in a single gesture. The average time difference between a pen-up and pen-down for the double-stroke delete gesture was found to be around 250 ms from experiments based on 4 users with a HP5455 iPAQ PDA. In our implementation, any stroke made within 300 ms of one stroke would buffer the first stroke and take in the current stroke as the latter part of a double stroke. Any stroke made after 300 ms from a previous stroke would be considered to be a separate single stroke gesture.

Using these concepts a pen gesture with two strokes that intersect each other on the screen is considered a uni-modal command to delete an object from the screen. Figure 2 (right) shows an example of this double-stroke pen gesture. However, users may also employ a multimodal command combining speech and pen tap to delete objects.

2.3. Pen Taps for Menus and Dialog Boxes

In addition to pen taps and gestures which were combined with speech input we also implemented an input modality based on pen alone which employs menus and dialog boxes. This type of input represents the standard modality for such handheld computers. It was implemented here to allow users to interact with the applications without using speech input which is not appropriate in some cases. Figure 3 shows an example of the screen depicting the use of menus (left) and a photograph of the screen depicting a dialog box for creating objects (right). Dialog boxes are employed to specify input arguments for the commands, such as color, orientation, etc.

Figure 3. Example screens for using menus selected with the pen.

3. APPLICATIONS AND PRELIMINARY EVALUATION

We implemented the new functions in our original application for creating diagrams of the accident scenes [14]. Two other applications were developed for interacting with maps on PDAs by speech and pen gestures as presented in Figure 4. The first application (displayed on the left hand side in Figure 4) aims at providing users with city maps for real estate applications in which users can create, move, and delete symbols on the maps representing interesting properties. Such symbols can encode the price range of the properties by using colours. The second application (displayed on the right hand side in Figure 4) aims at providing tourists with vacation maps on which they can place various symbols and notes regarding their vacation trips. The speech engines employed in this work were developed by HandHeld Speech, LLC (http://www.handheldspeech.com). The embedded ASR engine is a large vocabulary speech recognition engine built by the above company for Windows CE or Windows Mobile. The ASR engine works in a speaker dependent mode and requires that users enrol first by providing a few utterances for adapting the acoustic models. The acoustic models of the speech recognition engine can also be adapted during the use of the application by capturing the user's speech as shown in Figure 5. The speech engines also contain a simple text-to-speech to provide the user with supplementary speech feedback.

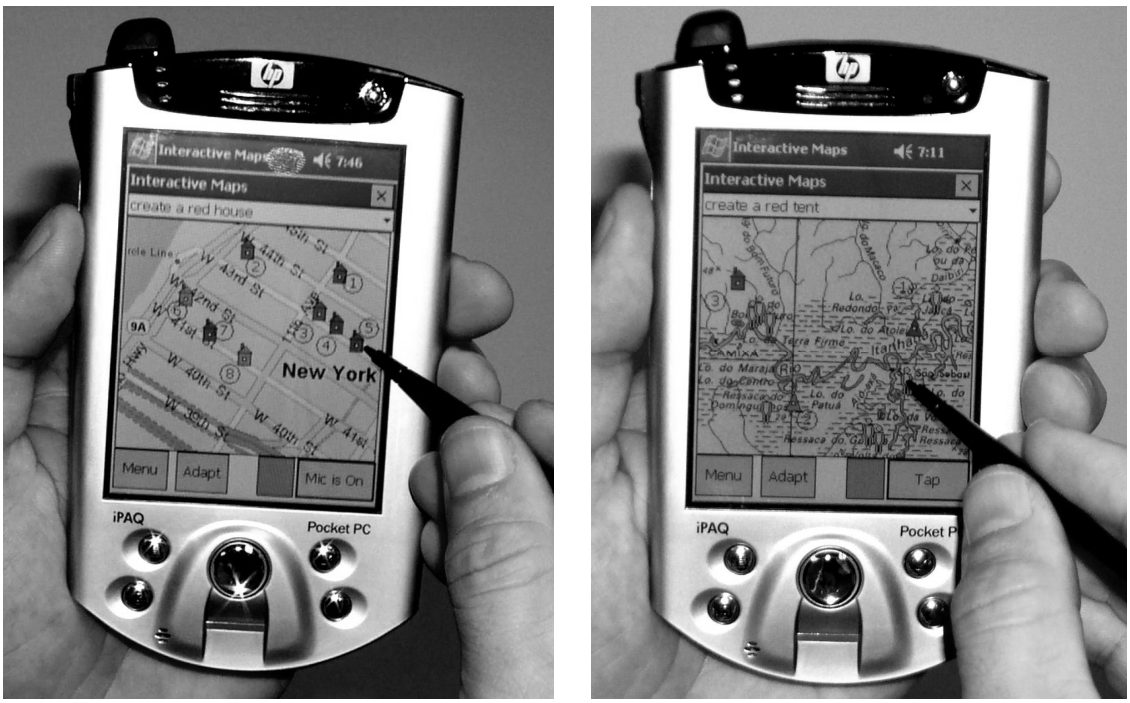

Figure 4. Example screens of two map-based applications.

In a multimodal interaction (speech and pen tap) employed by a user to move, for example, a car icon on the display from one position to another position, the user needs to say "Move this car" and tap on the desired car and tap again the screen at the new destination location. A similar function can also be implemented by employing a unimodal (pen) interaction consisting of two taps to open the menu and specify the action and the other two taps to specify the car and the new location position. It is obvious that the multimodal interaction is not only more natural but also faster.

A preliminary evaluation of the three applications was performed by four users. The purpose of this preliminary evaluation was to compare the duration and naturalness of various unimodal (speech or pen) and multimodal constructs. The flexibility of the interface derives from the possibility of allowing users to choose at any time the type of interaction (unimodal or multimodal) they prefer. For example, if a user is in a

noisy environment, he/she may choose the unimodal pen interaction. In a quiet environment, the preferred choice would be multimodal.

Figure 5. Example screen of adapting the acoustic speech models to the user.

The preliminary evaluation involved the creation of a given accident scene by employing speech and/or pen taps and gestures. Table 1 shows a comparison between different modalities employed to create the same accident scene. The table displays the type of action on the left column and the number of pen taps or strokes necessary for unimodal or multimodal (speech and pen) interactions. The type o interaction is printed in the last column. The interaction can be pure speech (first action to create a road), pure pen gesture (last action to delete an object) or multimodal (speech and gesture). The number of pen taps or strokes is much smaller in multimodal interaction. For example, to create an icon of a house on the display a user requires ten taps in a unimodal pen interaction and only one stroke in a multimodal (speech and pen) interaction. Also the time to complete an accident diagram is shorter on average when users employ multimodal interaction. This shows that multimodal interactions are not only more natural but also faster than unimodal interactions.

Table 1. Comparative number of taps or strokes in various multimodal and unimodal interactions

Action	No. of pen taps (T) or Strokes (S)		Interaction
	Unimodal	Multimodal	
Create Road	6T	0T	Pure Speech
Create Object	10T	1S	Multimodal
Create Label	6T	1T	Multimodal
Rotate Object	4T	1S	Multimodal
Move Object	4T	2T	Multimodal
Delete Object	4T	1T	Multimodal
Change Road	6T	0T	Pure Speech
2-stroke delete	2 strokes	-	Pure Gesture

4. CONCLUSIONS

This paper presents improvements to our initial approach to speech and pen interaction on handheld computers and devices [14]. The improvements allow users to interact in a more flexible way by employing speech and/or pen taps and gestures in unimodal or multimodal fashion. The choice of the type of interaction is left to the user at any time. Depending on the circumstances, the same action is more appropriate to be performed in either a unimodal or multimodal fashion. An informal evaluation using four subjects shows a more natural, flexible and easy-to-use interface than our previous version. More evaluations and usability studies are necessary to fully understand the advantages and limitations of such multimodal interfaces.

5. ACKNOWLEDGEMENTS

This research was supported by the U.S. National Science Foundation under the Knowledge and Distributed Intelligence project, Creativity Extension Grant NSF IIS-98-72995.

REFERENCES

1. BOLT R., Put-that-there: Voice and gesture at the graphics interface, Computer Graphics, 14(3), pp. 262-270, 1980.
2. OVIATT S.L., DEANGELI A., and KUHN K., Integration and Synchronization of Input Modes During Multimodal Human Computer Interaction, Proc. Conference on Human Factors in Computing Systems, Atlanta, GA, ACM Press, pp. 415-422, 1997.
3. COHEN P. R., *et al*., QuickSet: Multimodal Interaction for Distributed Applications, Proc. of the 5th International Multimedia Conference, pp. 31-40, ACM Press, 1997.
4. JOHNSTON M., Multimodal language processing, Proc. of ICSLP, Sydney, Australia, pp. 2343-2346, 1998.
5. SHARMA R., PAVLOVIC V., and HUANG T., Toward multimodal human-computer interface, Proc. IEEE, **Vol. 86**, no. 5, pp. 853-869, 1998.
6. OVIATT S. L. *et al*., Designing the User Interface for Multimodal Speech and Pen-Based Gesture Applications: State-of-the-Art Systems and Future Research Directions. Human-Computer Interaction, **Vol.15**, pp. 263-322, 2000.
7. OVIATT S., Multimodal Interactive Maps: Designing for Human Performance, Human-Computer Interaction, **Vol.12**, pp. 93-129, 1997.
8. DUSAN S., and FLANAGAN J. "Adaptive Dialog Based Upon Multimodal Language Acquisition," Proc. of the 4th IEEE International Conference on Multimodal Interfaces, Pittsburgh, Pennsylvania, USA, pp. 135-140, 2002.
9. DEN OS E., DE KONING N., JONGEBLOED H., and BOVES L., Usability of a Speech Centric Multimodal Directory Assistance Service, Proc. of the Intern. Workshop on Information Presentation and Natural Multimodal Dialogue, Verona, Italy, pp. 65-69, 2001.
10. HUANG X. *et al*., MIPAD: A Multimodal Interaction Prototype, Proc. of the ICASSP 2001.
11. HASTIE H. W., JOHNSTON M., and EHLEN P., Context-Sensitive Help for Multimodal Dialogue, Proc. of the 4th IEEE International Conference on Multimodal Interfaces, Pittsburgh, PA, USA, pp. 93-98, 2002.
12. ALMEIDA L., *et al*., "Implementing and Evaluating a Multimodal Tourist Guide", Intern. CLASS Workshop on Natural, Intelligent and Effective Interaction in Multimodal Dialogue Systems, Copenhagen, Denmark, pp.1-7, 2002.
13. COMERFORD L., FRANK D., GOPALAKRISHNAN P., GOPINATH R., and SEDIVY J., The IBM Personal Speech Assistant, Proc. of the ICASSP 2001.
14. DUSAN S., GADBOIS G. J., and FLANAGAN J., "Multimodal Interaction on PDA's Integrating Speech and Pen Inputs," Proc. of Eurospeech 2003, Geneva, Switzerland, 2003.
15. COHEN P. R., COULSTON R., and KROUT K. "Multimodal Interaction During Multiparty Dialogues: Initial Results," Proc. of the 4th IEEE International Conference on Multimodal Interfaces, Pittsburgh, PA, USA, pp. 448-453, 2002.