

# Multimodal Interaction on PDA's Integrating Speech and Pen Inputs

*Sorin Dusan<sup>a</sup>, Gregory J. Gadbois<sup>b</sup>, and James Flanagan<sup>a</sup>*

<sup>a</sup>Center for Advanced Information Processing, Rutgers University, NJ, U.S.A.

<sup>b</sup>HandHeld Speech LLC, Amesbury, MA, U.S.A.

sdusan@caip.rutgers.edu, greg@handheldspeech.com, jlf@caip.rutgers.edu

## Abstract

Recent efforts in the field of mobile computing are directed toward speech-enabling portable computers. This paper presents a method of multimodal interaction and an application which integrates speech and pen on mobile computers. The application is designed for documenting traffic accident diagrams by police. The novelty of this application is due to a) its method of fusing the speech and pen inputs, and b) its fully embedded speech engine. Preliminary experiments showed flexibility, versatility and increased naturalness and user satisfaction during multimodal interaction.

## 1. Introduction

An emerging research direction in the field of human-computer interaction (HCI) aims at making the interaction more natural and efficient by integrating multiple input-output modalities into the interface. Speech is one of the potential candidates that could contribute to both naturalness and efficiency. Mouse and pen gestures can also increase these attributes of interaction if used in combination with speech or when speech is not efficient, for example to specify spatial positions or to refer to spatial objects. The way these two input modalities are integrated dictates the level of freedom offered to users and, finally, influences the efficiency of the interaction and the user satisfaction.

Various methods of integration of multiple modalities have been proposed since the original system "Put that there" developed by Bolt [1]. More recent approaches have been published by Oviatt et al. [2], Cohen et al. [3], Johnston [4], Sharma et al. [5], Dusan and Flanagan [6]. A comprehensive review of methods for integrating speech and pen inputs has been published by Oviatt et al. [7]. One of the most common multimodal applications is based on interactive maps, for locating points of interest such as restaurants, museums, real estate, or for providing driving directions. A comprehensive study on multimodal interactive maps designed for enhancing human performance was published by Oviatt [8].

During the last few years, increased research efforts have been focused on adding multimodal capabilities to mobile computers, such as personal digital assistants (PDA's) or tablet PC's. Since these portable devices do not have physical keyboards but rather small soft input panels (SIP) and pens, a natural input addition to them would be speech. Integrating state-of-the-art speech recognition and synthesis engines into these portable devices is difficult because of the limited computing power and small memory capacity. An alternative is to speech-enable these portable devices by performing the recognition and synthesis on wirelessly connected, more powerful servers. Some of the recent approaches have been

published by Den Os et al. [9], Huang et al. [10], Johnston et al. [11]. An embedded-speech approach has been published by Comerford et al. [12]. This application uses speech as primary mode of interaction and is based on a PDA which contains an embedded speech engine that performs speech synthesis and speech recognition based on a vocabulary of up to 500 active words.

In this paper we present a multimodal interaction method and an application on PDA's using an embedded speech engine. The speech engine performs speech synthesis and automatic speech recognition for vocabularies of over 10,000 active words. This application flexibly integrates speech and pen inputs and provides graphics, text, and synthesized speech as output modalities.

The paper is organized as follows: Section 2 outlines details of multimodal interaction with portable computers and section 3 presents an application implementing multimodal interaction. Preliminary tests and evaluations are described in section 4, whereas section 5 concludes with discussion and future research directions.

## 2. Multimodal Interaction

Multimodal interaction, when available, is strongly preferred over unimodal interactions [8]. This applies also to portable computing devices which present some limitations, as compared to regular desktop computers or laptops. This section presents a method of multimodal interaction with these devices based on pen and speech inputs, and graphics, text and speech output.

### 2.1. Pen and speech inputs

Since portable computing devices, such as PDA's and Tablet PC's, do not usually come with common hardware input devices, such as mouse and keyboard, the input functions are accomplished by using touch screens and pens. Miniature software keyboards are displayed on the screen and used by tapping the characters with the stylus. The stylus serves also for pointing and replaces some of the functions of the mouse. These portable computers typically run Windows CE which is a special version of the Windows operating systems. A common implementation of this operating system is Pocket PC. Because of lack of a mouse these portable devices usually do not have a cursor on the screen. This limits pointing to tapping on the screen, which is equivalent to mouse clicking. Mouse moving events can also be generated by moving the pen while keeping contact with the touch screen.

Integrating speech input for these portable computers is possible since the hardware requirements already exist in most of these devices. The challenge is the software implementation of a speech engine for recognition and synthesis. As

mentioned in the introduction, some solutions for this problem rely on wirelessly connected servers that carry the computationally expensive tasks of automatic speech recognition (ASR) and speech synthesis. However, this solution comes with some drawbacks, for example, the geographic limitation to places where the wireless connection to the server is available.

Another solution is to build an embedded speech engine with a small footprint capable of fitting the limited computing and memory resources of these portable computers. Such an implementation was presented in [12]. In this paper we introduce another embedded speech engine, featuring a speech recognizer capable of large vocabulary continuous speech recognition, [13]. In general, vocabulary size is not an issue with this engine and it can commonly perform speech recognition for over 10,000 active words. Microprocessor requirements scale logarithmically with vocabulary size. Once you reach large vocabulary, doubling the size of the vocabulary has almost no effect on the microprocessor usage. Memory requirements scale linearly with vocabulary, and become the limiting factor for truly large vocabulary on the embedded hardware. The biggest limitation on this engine is that it currently has no language model. It is a pure phonetic recognizer and as such, it cannot address an open dictation problem. However, for a command and control application this is not a problem. This speech recognizer requires a very short enrollment for adaptation.

## 2.2. Graphics and speech outputs

Multimodal interaction with computers requires not only multimodal inputs but also multimodal outputs, to provide information to users and feedback for the input commands. PDA's and Tablet PC's are equipped with graphical screens so that graphics and text can be displayed to the users. This display represents the primary output modality.

Another output modality on these devices is represented by the audio output. Most portable computers have built-in hardware capabilities to play audio on tiny speakers. Hence, speech output is also possible by employing a text-to-speech (TTS) software. In our research we employed a very simple TTS implementation embedded in the portable device. TTS engines run a gamut from the very small (sounding very machine like) to the largest state-of-the-art systems that sound very natural. The TTS engine we employed, [13], is particularly small (and sounds machine-like). For a small set of prompts, we use simple wave file recordings. For example, to repeat back a digit string we have the digits recorded singly and we play back a catenation of the corresponding wave files. We use the TTS engine where it is not feasible to use a small set of wave file. The TTS engine runs on phonetic spellings, the same spellings used by the recognition engine. Text is looked up word by word in the phonetic dictionary (shared with the recognition engine) or if the word is out of vocabulary, it falls back to rule based pronunciation guessing.

## 2.3. Multimodal fusion

A multimodal interface should offer the possibility to use various input modalities not only separately but also simultaneously and complementarily. Using multimodal inputs in a complementary way increases the naturalness of the interface. During a dialogue act, information about required actions or objects' names and attributes is easier to provide by

speaking, whereas spatial information, such as objects' positions in space, is easier to provide by pointing.

In [6, 14] we presented a method of fusing speech and mouse inputs for multimodal applications on desktop or laptop computers. We will briefly summarize this method here in order to introduce its principle and to emphasize the differences imposed by the handheld devices. This fusion method is based on timestamps associated with each individual modality. The timestamps corresponding to the beginning of each spoken word were used to identify the cursor position on the screen when those words occurred, by comparing them with the timestamps corresponding to the mouse moving WM\_MOUSEMOVE Windows events. This method does not require mouse clicking. Locating the **X** and **Y** coordinates of the cursor corresponding to each word offered a solution for solving deictic references to objects such as "this" or "that" or to positions such as "here" or "there". Figure 1 shows a schematic time diagram describing this principle for an utterance containing the deictic references "this" and "here".

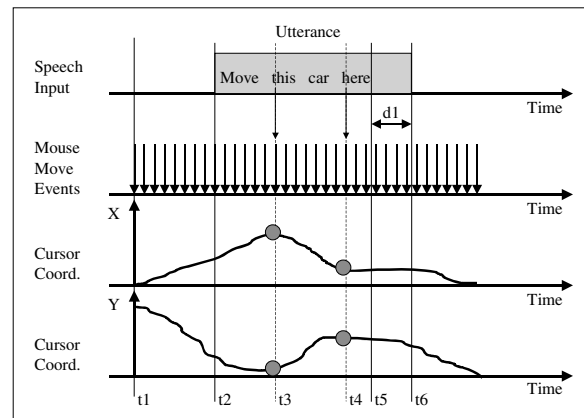


Figure 1: Schematic time diagram for the fusion method based on timestamps.

As seen in this figure, the mouse started to move at  $t_1$ , prior to the beginning of the utterance  $t_2$ . The mouse move events are represented in the second plot as arrows. The end of the utterance was detected by the ASR based on the silence interval  $d_1$ , between  $t_5$  and  $t_6$ . Then the ASR provided the timestamps corresponding to the beginning of the words "this" and "here". These timestamps were used to identify the mouse move events at  $t_3$  and  $t_4$  and the corresponding **X** and **Y** coordinates of the cursor.

There is one problem in implementing this fusion method on portable devices and this is because handheld devices do not have a mouse and do not display a permanent cursor on the screen. The Windows CE operating system treats the pen tap on the touch screen as a WM\_LBUTTONDOWN Windows message (equivalent to a mouse click). Thus, the fusion method presented above needs to be adapted to work with the WM\_LBUTTONDOWN messages and the WM\_MOUSEMOVE messages. We 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 fusion method using only the WM\_LBUTTONDOWN messages.

Another distinction from our previous fusion method is motivated by recent user studies that showed that most of the time 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 speech, this is not true all of the time and not for all users. Thus, for some users speech precedes gestures with a time interval between 1 and 2 seconds [15]. These observations motivated us to implement a more flexible technique for fusing speech and pen inputs.

This new fusion method is especially suitable for handheld devices but not exclusively. Figure 2 presents a schematic time diagram depicting the principle of the new method for fusing speech and pen inputs.

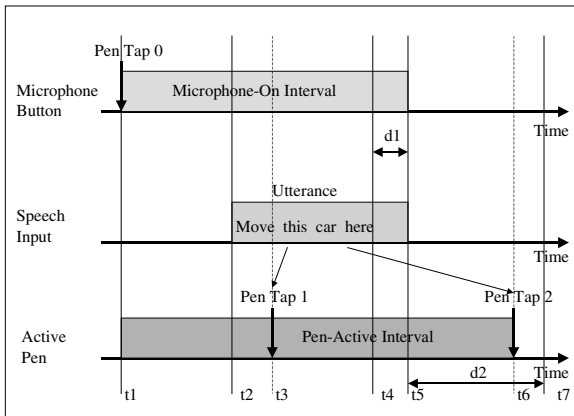


Figure 2: Schematic time diagram for the handheld method of fusing speech and pen inputs.

The multimodal command starts with the pen tapping on the microphone button at time  $t_1$ . This initial pen tap (Pen Tap 0) switches the microphone to the ON position, which remains in this state until an utterance is issued or until the pen taps again the microphone button and turns it into the OFF position. The new method allows that after  $t_1$ , either the gesture or the speech can begin the multimodal construction. In the example shown in Figure 2, speech starts at  $t_2$  and precedes the pen gesture that occurs first at  $t_3$  (Pen Tap 1) for the deictic reference “this”. Note that Pen Tap 1 precedes in this example the occurrence of the word “this”. However, the user is not required to tightly synchronize the pen gestures with the deictic references, and thus these gestures can precede, overlap or succeed the deictic words. In order to implement this flexibility we built a circular 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. In the example of Fig. 2, the utterance contains two deictic references but only one pen tap, at  $t_3$ , has been received by the end of the utterance  $t_4$  or by the time  $t_5$  when the ASR detects its end. If the utterance, after interpretation, indicates that it contains two deictic references, a timer then starts at  $t_5$

and waits within a  $d_2$  interval of maximum 3 seconds for the second pen tap, which in the example occurs at  $t_6$ . At  $t_5$  the TTS speaks back “Tap the screen”. If the second tap were not received within the  $d_2$  interval, then the whole command would have been discharged at  $t_7$ . If the utterance contains only one deictic reference, for example as in “Delete this truck”, only one pen tap is required for the deictic resolution. Again, this pen gesture can precede, overlap or succeed the deictic word. If two taps are received instead of one, only the second one is processed. The new fusion method described here provides users with more flexibility in composing multimodal constructions.

### 3. Multimodal Application

We implemented the above method of multimodal interaction on an iPAQ PDA as an application for documenting traffic accident reports by police. The application was written using the Embedded Visual C++ programming language.

As a result of a traffic accident, police have to file a report documenting various related pieces of information. Among them are the participants involved in the accident (pedestrians or drivers of vehicles), their vehicles, their positions or vehicle positions at the time of the accident, and whether or not they were indicating changes of directions. This task is usually done at the site of the accident by a police officer based on the visual observations and statements of the participants and witnesses of the accident. A simple photograph of the scene of the accident is not useful in many cases if the vehicles have been moved prior to the police arrival. The police officer has to draw on paper the accident diagram corresponding to the moment of the accident, and this depends on his or her drawing skills. The task of this application can be carried on by interacting multimodally with the handheld device by fusing speech and pen inputs.

Using speech commands the user can create various intersection or road diagrams, for example, by saying “Create a cross intersection” or “Create a T intersection”. Then, using speech and pen inputs, the user can place in the desired location of the screen various icons symbolizing pedestrians, cars, trucks, busses, bicycles or motorcycles. The user can attribute colors to these vehicles, for example, by saying “Create a red car” or “Create a white truck”. Later these icons can be moved to different locations or rotated to different orientations, for example by saying “Move this car here” and tapping the pen on the car and then on the screen to show the new location, or by saying “Rotate this bicycle to 90 degrees the left” and tapping on the corresponding bicycle. Another feature is the possibility to display turning indications or lights of the vehicles, for example by saying “Indicate turn left for this car” or “Indicate turn right for this bicycle”. When the accident diagram is completed, the user can assign and display a label indicating a number for each participant at the accident. At run-time, users can adapt their voice profiles using a combo box and an Adapt button if a command was misrecognized. The application is based on a rule grammar containing 10 rules. In the office or in the police car, the created accident diagram can be transferred, using the Microsoft Active Sync software, into a computer by placing the PDA into its cradle.

Figure 3 presents a picture of the PDA application for documenting a simulated accident diagram containing an intersection with pedestrians, cars, trucks, bicycles and labels.



Figure 3: Multimodal PDA application based on speech and pen inputs.

#### 4. Preliminary Experiments

We conducted preliminary experiments with 4 users, each of them completing the task of creating two different accident diagrams by combining speech and pen inputs. Each user adapted the ASR by creating his voice profile during a 5 minutes enrolment. Then each user was familiarized with the allowed grammar rules and utterances. Two different accident diagrams printed on paper were given to the users at the beginning of the experiments and the users were asked to reproduce them on the PDA by interacting multimodally. After completing the task the users were asked to express their satisfaction regarding the multimodal interaction in terms of naturalness (N) and of ASR accuracy (A) on a scale from 1 (low) to 10 (high). The results were: User1(N)=10, User2(N)=9, User3(N)=8, User4(N)=9, User1(A)=9, User2(A)=8, User3(A)=9, User4(A)=8.

#### 5. Conclusions

Users benefit from multimodal interaction based on speech and pen inputs mainly in applications dealing with spatial domains, such as maps. There are several reasons why this is true, including less complex command constructions, faster completion of tasks, fewer errors and disfluencies, and almost unanimous user preference for multimodal interaction [8]. New emerging research aims at speech-enabling portable computing devices such as PDA's and Tablet PC's. In this context, we developed a new multimodal integration method and implemented it in a PDA application for documenting traffic accidents. Preliminary experiments show an increase in naturalness and efficiency in operating the application multimodally. Future work will focus on building and evaluating other multimodal applications on handheld devices. A short video demo of this application and further

developments and information can be found on the web at: <http://www.caip.rutgers.edu/speech/multimodal/>.

#### 6. 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.

#### 7. 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, 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] Dusan, S., and Flanagan, J., "A System for Multimodal Dialogue and Language Acquisition", In *Speech Technology and Human-Computer Dialogue*, Romanian Academic Publishers, Bucharest, Romania, April 2003.
- [7] 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.
- [8] Oviatt, S., "Multimodal Interactive Maps: Designing for Human Performance", *Human-Computer Interaction*, Vol.12, pp. 93-129, 1997.
- [9] Den Os, E., De Koning, N., Jongbloed, 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] Johnston, M. et al., "MATCH: An Architecture for Multimodal Dialog Systems", *Proc. of ACL-02*, Philadelphia, 2002.
- [12] Comerford, L., Frank, D., Gopalakrishnan, P., Gopinath, R., and Sedivy, J., "The IBM Personal Speech Assistant", *Proc. of the ICASSP 2001*.
- [13] Gadbois, G., HandHeld Speech, LLC, <http://www.handheldspeech.com>
- [14] 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.
- [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, Pennsylvania, USA, pp. 448-453, 2002.