AN ADAPTIVE MULTIMODAL INTERFACE FOR WIRELESS APPLICATIONS

Jie Yang, William Holtz, Weiyi Yang, Minh Tue Vo

Interactive Systems Laboratory
Carnegie Mellon University
Pittsburgh, PA 15213 USA
{yang+, wmholtz, wyyang, tue}@cs.cmu.edu

Abstract

In this paper, we present an adaptive multimodal interface for wireless computing applications. The system optimizes its performance by dynamically selecting its network service based on cost and performance requirements. We describe a prototype system and its application to demonstrate the proposed concept.

1 Introduction

The mobile computing technology has made it possible for portable computers to access information from any location. In order to maximize the effectiveness of systems in mobile computing environments, interface design must be matched with user tasks [1]. The effectiveness of multimodal human-computer interaction has been demonstrated by many researchers for different applications. It will also be natural to employ multimodal interfaces in mobile computing devices. Mobile devices connected to a wireless network, however, suffer wide variations in network conditions. Coping with the network uncertainty requires the ability to share remote and local resources at varying degrees of fidelity [2]. In this paper, we present an adaptive multimodal interface based on a dual server structure for wireless computing applications. The underlying idea is to dynamically determine the network usage based on cost and performance requirements. The input signals are either handled by the local multimodal server, or by the remote multimodal server, or partially by the local multimodal server and partially by the remote multimodal server.

2 Dual Server Structure

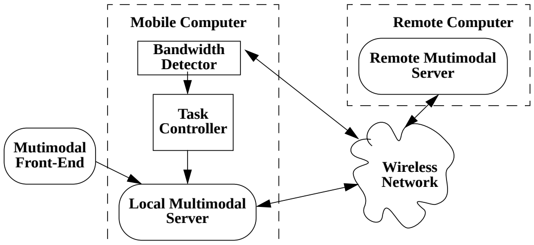

Figure 1. The dual server structure

Mobile computers can be weakly interconnected by low-speed wireless networks, or disconnected for some reasons. When the computer is connected to network, it can share a variety of resources via network. When the computer is disconnected from network, it should be able to perform it's task at least at a minimum level. For example, a car navigation system should be able to respond to a query at any time. When the computer is connected to a network, the system can send the query to a computer with more computing power and a larger, more up to date database and get a quicker and more accurate response. When the computer is disconnected from the network, such as the car driving through a tunnel, the system can still answer the query by searching the database in local storage. The same idea can be applied to a multimodal interface. For a multimodal interface, the task can be allocated between the local and remote server based on the network connection and computational cost. It is desirable that the system has the ability to dynamically select its network service based on cost and performance requirements. This can be achieved by a dual server structure as shown in Figure 1. The multimodal inputs can be processed locally, or remotely, or partially local and partially remote. Each server has a similar architecture. The server can perform recognition for the modalities of speech, handwriting, and pen gesture, and interpret the recognition results. From the communication point of view, we only need to handle speech and pen inputs for modalities of speech, handwriting and gesture.

3 A Prototype System

The primary obstacles in creating a dual-server system on a wireless network are creating multimodal servers which can process a fraction of a multimodal interaction, creating network connections between the servers, and determining the optimal use of limited local processing power under a range of bandwidths. In order to demonstrate the proposed concept, we have developed a prototype system. We use an IBM ThinkPad 701C (Intel 486 CPU, 75MHz, 24 Megabytes memory, 512 Megabytes hard disk) with a wavelan card as a mobile computing device. The ThinkPad is connected to another PC (Pentium II dual processor, 266MHz, 512 Megabytes memory) via 915MHz wireless network. We have installed multimodal servers on both computers.

The system was designed to tolerate change in network throughput at any time, though, in the current experiment, we assume that the bandwidth of the network connection would remain constant for the duration of sending multimodal interaction data to the remote server and receiving back results. The system can quickly adapt to different network traffic by scheduling the task to local and remote servers. All communications between the servers is done through TCP/IP sockets, with open connections only existing during data transfers in order to reduce the chance of network failure occurring while a connection exists.

To adapt the system to different network conditions, the system has to be able to adjust the amount of data it is sending through the network. The multimodal server needs to handle two types of data: pen and voice input. For the pen input, we only need to deal with coordinates of sampling points. This produces a relative small amount of data that can be further reduced through simple techniques such as run-length-encoding to the extent that network bandwidth doesn't matter. Although an audio stream for speech recognition at 32 kByte/s poses no problem for a regular Ethernet connection, the situation is quite different in a wireless communication network where a low bandwidth connection has to be shared among multiple applications and even multiple users (broadcast medium). But since there is a lot of redundancy in audio data for a speech recognizer, the system has enough room to adapt to different network conditions. Our experiments showed that the feature data for speech recognition could be less than 7% of the original audio data size.

By analyzing the speech recognition process we have partitioned its feature extraction process into seven levels for which the processing could be scheduled between servers. At level zero all processing is done locally. Level one does all feature processing locally and the search remotely. Level two additionally moves matrix size reduction computation to the remote server. Level three additionally moves matrix multiplication computation to the remote server. Level four additionally moves the adjacent frame combination processing to the remove sever. Level five additionally moves the normalization computation to the remote server. Level six additionally moves the fast Fourier transformation to the remote server, so that no computation is done locally and the raw audio is transferred to the remote server.

The primary variable in the system is the current bandwidth of the network connection. It is a safe assumption that the processing power of the local and remote servers will remain relatively constant. In order to calculate which processing split is the best for the current bandwidth it is necessary to know the processing power of the local and remote servers. We hope to find a relationship between processing time and network traffic (data size and bandwidth). For a specific application, the range within data sizes will vary is known, and the range of data sizes for most applications is small. Therefore a linear approximation of the data should be accurate as long as the data size stays within the expected range. Our experiments have shown that the following function could characterize the relationship between processing time and network traffic:

$$f_{level}(S, B) = K_0 + K_1 S + K_2 \frac{S}{B} \tag{1}$$

where f_{level} is the processing time by selecting network service at $leve$, $level = 0, 1, \ldots, 6$; S is the data size in bytes and B is the network bandwidth in bytes/s. K_0, K_1, and K_2 are constant.

The prototype system has been tested by several applications. An example is the QuickDoc [3] system. The task is for a doctor to go through a series of images such as X-rays or computer-aided tomography scans, quickly identify an anomalous area, label the area with the name of a disease or condition, and attach relevant comments using voice and pen. The end product is an HTML report that summarizes the doctor's findings in a compact table listing the annotated images, the corresponding preliminary diagnoses, and automatically generated hot-links to relevant sites based on the diagnoses.

References

[1] A. Smailagic and D.P. Siewiorek, “Modalities of interaction with CMU wearable computers," IEEE Personal Communications, vol.3, no.1, pp. 14-25, 1996

[2] M. Satyanarayanan, “Mobile information access," IEEE Personal Communications, vol.3, no.1, pp. 26-33, 1996.

[3] A. Waibel, B. Suhm, M.T. Vo, and J. Yang, “Multimodal interfaces for multimedia information agents," Proc. of ICASSP'97.