Poly-sensing Environment for Poetic/Informational IT System

Shenglin Yang and Ingrid Verbauwhede
Electrical Engineering Department, University of California-Los Angeles, Los Angeles, CA 90095

Fabian Winkler and Bill Seaman
Department of Design | Media Art, University of California-Los Angeles, Los Angeles, CA 90095

Abstract

In this paper, an inter-disciplinary research project is presented between researches for electrical engineering and digital media arts. Our goal is to create a poly-sensing environment to enable the appurtenant extension of the human senses. An authoring tool enables this sense data to be linked to a virtual environment.

This paper describes the overall goal of the project. Secondly it shows some first step results obtained by a wireless speech recognition environment linked to a virtual media space.

1. Introduction

Advances in sub-micron and nano-scale semiconductor technologies allow the integration of multiple heterogeneous sensors on one "system-on-a-chip". The unique aspect of this technology will be the collection of information from the parsing of an integrated "collaboration" between different selected micro-scale sensing devices. It allows to create a tool with a flexible and adaptive means to focus the "attention" of these multiple-sensing devices, in recombinant groups of intercommunicating distributed fields.

The goal is to generate a robust authoring system that will enable the focusing of the poly-sensing environment and subsequently to link the sense data to other controlled computer-triggered events and behaviors. The environment will enable inter-communication between devices embedded in many kinds of equipment and places, exploring the needs for extremely diverse wireless connectivity. The system will seek to be self-observant and thus to predict reliability and performance. Thus the goal is to develop a scalable system with enormous potential as well as to ensure vigorous and dependable operation. The poly-sensing environment will seek to balance hardware and software functionality, exploring flexible computation and communication in its system architecture. The technology will also include the potential for encryption of the multiple streams of sense data.

Sensor networks are of growing importance. It is being proposed for environmental and habitat monitoring [1], for office climate control and energy monitoring [9]. MIT's Project Oxygen Alliance is developing a new form of computing and communication [10]. It is an integrated vision and speech system using cameras and microphone arrays to respond to a combination of pointing gestures and spoken commands. Researchers in Hewlett-Packard Laboratories describe the optimization of a signal processing front-end for a distributed speech recognition system with the goal of reducing power consumption [4].

2. System Overview

A general view of the poly-sensing environment is shown in Figure 1. The interaction with the environment is illustrated in Figure 2. This environment has seven inter-functioning levels. We start at the sensor ends, where the raw data is collected and we end at level 7, the networked virtual reality/poly sensing spaces. For each level, the different challenges and design issues will be discussed.

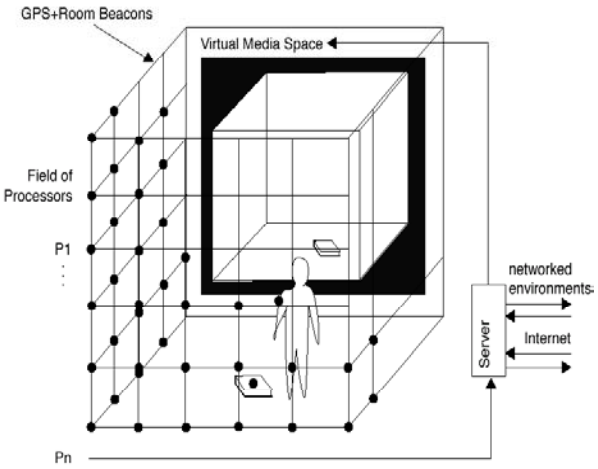

Fig. 1. The poly-sensing environment.

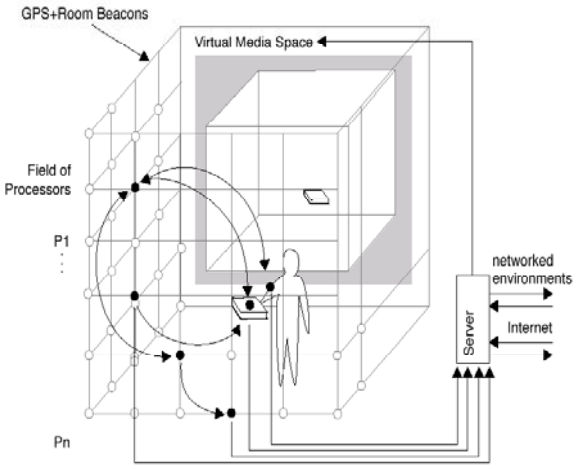

Fig. 2. Interaction with the poly-sensing environment.

The first three levels describe the node, which is illustrated in Figure 3. This node will be an integrated embedded system on a chip.

Level 1: Poly-sensing device

The first level is the collection of raw data by the individual sensing devices. These sensing devices will range from cheap infrared or temperature sensors to more expensive video cameras. The idea is to provide sensory support for each of the human senses: visual, hearing, smell, taste, and touch, extended with non-human "sensing": chemical (extension of smell and taste), infrared (extension of vision), ultra-sonic (extension of hearing) and so on.

The sensor nodes don't have to be all the same. For instance, one could sprinkle cheap sensor nodes, as described in [1] or in the Piconet project [2], onto the walls of the room, and combine them with a few more expensive video nodes.

The idea is that the authoring tool selectively turns on or off areas of interest and thus areas of sensors. In this way, video might be one set of nodes in the system (not on the chip but multiple cameras integrated with the chips) – or a new kind of "video" image might be constructed out of individual sensor data streams, given a particular perspective and series of single sources of light collection. Recognition may take place on a low level at the chip with high level processing of data at the server once data is collected from distributed chips. Many types of sensor can be involved in this system, e.g. sound, heat, pressure, chemical, etc.

Although building a SoC with different sensors is necessary, for our first prototype we use an existing platform to construct the first demonstration. The integration of sensors on one single chip is the next phase, once the environment has been built.

Level 2: Digital signal processing

The raw data from the sensors is collected and processed on the node by the processor. This processing consists of three phases. The first phase is the processing of data from the individual sensors. These are relatively simpler tasks such as correlating data between different consecutive measurements. For instance, the node could be instructed to check the variation in temperature or humidity. The second phase includes more high-end signal processing tasks. They are still at the level of one type of sensing. Examples are: image recognition, text recognition, gesture recognition, video, data,

sonar data etc. The third phase consists of signal processing between nodes and between different sensing devices. For instance, collaborative processing between nodes could indicate the movement of an object and is used to track objects. Another example is the collaboration between different senses: for instance, the increase in temperature combined with increase of humidity or pressure on a device could indicate the level of un-comfort of the person in the room.

Level 3: System on a chip

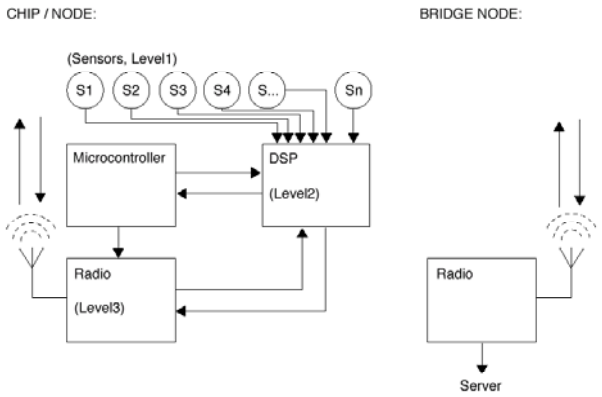

Fig. 3. Embedded "system on a chip".

One node, as shown in Figure 3, will also include a micro controller and a radio interface. The micro controller is the master on the embedded system. It collects the data on each chip and controls the power consumption, computation and communication resources. The processor defines what should be turned on/off - interaction of user with "Poly-Sensing Tool". It controls the articulation of "neighborhoods" or "fields" of "sensory" attention, defines what gets shared with other chips, detects "anomalies" in the streams and "broken sensors", and sends data via radio to the server. In addition, the processor is a possible site for aspects of encryption. The radio enables the node to communicate with the server, acting as a bridge node, and to communicate with neighboring nodes. These radios will need to be organized in an ad-hoc fashion.

At this level, it is not the intention to develop new radio protocols or new ad-hoc wireless communication protocols. These have been extensively studied within sensor research projects and commercial products, such as Bluetooth or 802.11 wireless LAN's.

Level 4: Server

The server is the bridge between the wireless sensor network and the wired back-bone. It has the function of defining location and relevant location of the distributed field of poly-sensors, as well as the position of each poly-sensing node. The serve will act as a parsing environment in which software can "make sense" of the varying streams, enable comparisons between the streams, and detect "anomalies" in the streams and "broken sensors". It can store relevant sense streams as a non-linear time-based database. The Poly-Sensing Tool in the server can define the virtual environment media-triggering and/or mapping of the observations made by the "fields of attention". In addition, the server will be a site for aspects of encryption.

Level 5: Filtered data for search engine

The fifth level describes the transition from the more hardware related levels to the pure software and virtual reality related levels. It contains the following functions: parsing data, sending appropriate data to high level processes, eliminating extraneous and/or non-changing data, and looking for error data. If/then statements make the sense data operative upon media elements and processes in the following virtual reality environment.

Level 6: Virtual reality environment

The virtual reality environment is a software environment to project the virtual reality back to the room. This environment can act on and react to sense data – visualizing different modes of sense data as virtual environmental maps, providing map of room which is a simultaneous view of sensor abstractions, and providing "images" of the room made by abstracting data from each of the sensors. It can also enable the

switching between a series of differing related "parallel" virtual environments, showing which parts of each field are active or passive, and sense data to control virtual media-elements defined by user with tool kit media/behavior matrix. In addition, this environment will be able to use sense data to initiate internet searches of "attached" functions – calling in related information and provide a space to view other networked environments as a distributed environment. Furthermore, the virtual reality environment can output the virtual environment to screen in the room and to networked virtual/actual spaces.

Level 7: Networked virtual reality environments and poly-sensing spaces

An exciting potential of the environment is its ability to function as a networked virtual environment, where international participants might simultaneously observe and/or explore/navigate the associated virtual environment as illustrated in Figure 4.

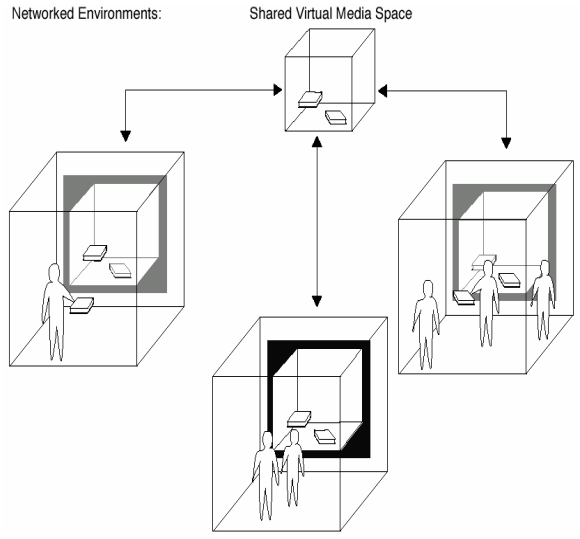

Fig. 4. Networked environments.

If a related set of objects populate both local and remote environments, comparisons of use might be made, empirical data generated and stored, or alternately, poetic relations shared. A copy of the media elements could be made available to both locations. Only difference information is sent over the internet, and both virtual environments are updated. Also central to the networked environments is the ability to intelligently search the internet for related data to be displayed in each poly-sensing environment, as designated by individual users and the matrix tool sets that enable one to define specific "if" "then" statements, i.e., if I lift this object, then go to the internet and find all other titles related to the author.

3. Prototype

Based on the seven-level design methodology discussed above, our prototype has been built.

3.1 Platform

The Compaq iPAQ H3835 personal digital assistant running Familiar Linux is chosen as our sensor node platform because it has reasonable battery life and the peripherals needed by our project. The iPAQ is a commercial handheld using a StrongARM SA-1110 processor and the memory for model H3835 is 64MB RAM. It includes a built-in microphone by which we can collect the speech data. Also it is possible to make the iPAQ more aware of what is going on around it with minimal feedback from its user by connecting other sensor devices to it via the serial port. A PC running WindowsXP is used as the server.

The idea for transmitting data is to integrate existing radios in the poly-sensing node prototype. We equipped the iPAQ with Orinoco Silver Wireless PC Card, choosing 802.11b as our communication protocol.

3.2 Speech processing for embedded iPAQ

Speech is one of the most important information sources in the poly-sensing environment. It can act as the connection between the user and the environment because speaking is the most natural way for people to communicate. Therefore speech recognition is implemented in our prototype.

In embedded platform, energy consumption is a critical issue. As a result, some regular speech recognition software, such as Sphinx II speech recognizer [7] and Hidden Markov Model Toolkit [8], cannot be ported directly in the embedded device. Considering the energy limitation of embedded platform, it is reasonable to move power-consuming tasks to the server and only deal with relatively simpler processing on the embedded device. This has been done with an overall energy minimization objective in mind.

The most direct way to implement the speech recognizer is to collect data in the embedded device and send them to the server where processing and recognizing are realized. It requires low capacity for embedded device but very high capacity for the wireless communication channel. Putting both the processing and recognizing in the embedded system and only transmitting the final recognized result to the server is most efficient according to minimizing the bandwidth requirement. However, this solution is not adoptable because of its high energy consumption.

To balance the computation requirement on embedded device and the bandwidth requirement on wireless communication channel, a trade-off approach is implemented in this project. The speech recognition task is divided into two parts: front-end feature extraction and recognition. The former step is relatively less complex. It is possible to implement this step on the embedded device and then send the compressed features to the server. The ETSI standard [3] is used for extracting feature vector of the speech. Then the complex recognition step is realized on server.

StrongARM SA-1110 processor, which is used in iPAQ, does not have on-chip floating-point processor, so all floating-point operations must be emulated in software. Algorithm described in ETSI distributed speech recognition standard contains lots of floating-point math functions (e.g. triangle function, logarithm, etc.) and it is usually computation intensive. Research showed 90% of the energy is consumed by floating-point emulation [4] and in the meantime, system execution time is intolerantly long. To meet the time and energy constraints of the embedded system, fixed-point arithmetic is introduced into the source code optimization.

In the algorithm for extracting speech features, pre-emphasis, framing, windowing, and computing power spectrum can be ported into fixed-point straight-forwardly. However, when computing Mel spectrum and Mel ceptrum, the multiplier of filter coefficient and the power spectrum needs to be implemented. The product often overflows the 32bit register. To solve this problem, C source code is substituted by assembly code. Another challenge is to take natural logarithm. In our project, a very efficient fixed-point algorithm with low complexity is used to solve this problem [4][6].

In the one-server/one-client prototype, the main work for server is to recognize the speech feature received from the field node and to response to it in the virtual media space. The block diagram of the prototype is shown in Figure 5.

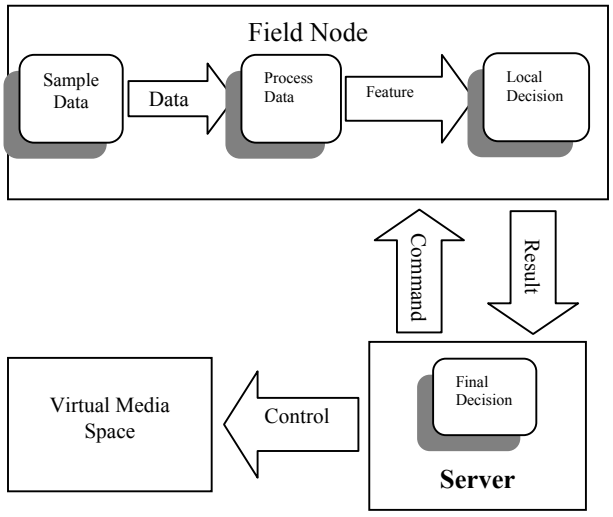

Fig. 5. Block diagram of prototype

The visualization part of our system was realized using Macromedia Director 8.5. The Director application reads commands from a text

file and based on these commands triggers media events (e.g. playing back a sound, opening a poem or image, or recombining various media on the digital canvas). The media files themselves are from futurist posters and performances allowing the user to constantly recombine them to new collages using spoken words.

4. Future Work

To implement object tracking or localizing, collaborative signal processing needs to be considered to get information about the object position as well as the direction of the information sources. To extend the ability of each sensor node, other types of sensors need to be added on it for combination of different data to get more accurate results on the server. In the meantime, energy efficient optimizations need to be explored from both software and hardware points of view.

Reference

[1] D. Estrin, L.Griod, G.Pottie, and M.Srivastava, "Instrumenting the world with wireless sensor networks," Proc. IEEE-ICASSP2001, Salt Lake City, Utah, May 2001.

[2] J. Rabaey, J. Ammer, J. da Silva, D. Patel, and S. Roundy, "Picoradio supports Ad Hoc Ultra-Low Power wireless networking," IEEE Computer Magazine, July 2000.

[3] "Speech processing, transmission and quality aspects; distributed speech recognition; front-end feature extraction algorithm; compression algorithms," ETSI Standard: ETSI ES 201 108 v1.1.2.

[4] B. W. Delaney, N. Jayant, M. Hans, T. Simunic, and A. Acquaviva, "A Low Power, Fixed Point, Front End Feature Extraction System for Distributed Speech Recognition," Proc. IEEE-ICASSP2002, Orlando, Florida, May 2002.

[5] "Writing Efficient C for ARM," Application Note 34, Jan 1998, ARM Inc.

[6] Jack W. Crenshaw, *Math Toolkit for Real-Time Programming*, CMP Books, Lawrence, Kansas, 2000.

[7] Carnegia Mellon University, "Sphinx-II automatic speech recognition system," http://www.speech.cs.cmu.edu/speech/.

[8] Speech Vision and Robotics Group of the Cambridge University, "Hidden Markov Model Toolkit," http://htk.eng.cam.ac.uk/.

[9] J. Rabaey, J. Ammer, T. Karalar, S. Li, B. Otis, M. Sheets, and T. Tuan, "PicoRadios for Wireless Sensor Networks: The Next Challenge in Ultra-Low-Power Design," Proceedings of the International Solid-State Circuits Conference, San Francisco, California, February, 2002.

[10] MIT Project OXYGEN, http://oxygen.lcs.mit.edu/index.html

[11] Project Webpage http://users.design.ucla.edu/~fwinkler/Poly_Sensing/

Acknowledgement

UCLA Chancellors Academic Border Crossing Fund

Langlois Foundation