13

ReachMedia: On-the-move Interaction with Everyday Objects

by

Assaf Feldman

B.S. Film & Computer Science, Tel-Aviv University (1999)

Submitted to the Program in Media Arts and Sciences,

School of Architecture and Planning,

in partial fulfillment of the requirements for the degree of

Master of Science in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNLOGY

September 2005

Author ______________________________
Program in Media Arts and Sciences
August 05, 2005

Certified by ______________________________
Pattie Maes
Associate Professor of Media Technology
MIT Media Laboratory

Accepted by ______________________________
Andrew B. Lippman
Chair, Departmental Committee on Graduate Students
Program in Media Arts and Sciences

ReachMedia: On-the-move Interaction with Everyday Objects

by

Assaf Feldman

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
on August 10, 2005, in partial fulfillment of the
requirements for the degree of
Master of Science in Media Arts and Sciences

Abstract

Mobile and wearable interfaces try to integrate digital information into our everyday experiences but usually require more attention than is appropriate and often fail to do so in a natural and socially acceptable way.

In this thesis we present "ReachMedia," a system for seamlessly providing just-in-time information for everyday objects. The system is built around a wireless wristband that detects objects that the user is interacting with and allows the use of gestures for interaction. Thus enables hands-and-eyes-free interfacing with relevant information using a unique combination of audio output and gestural input, allowing for socially acceptable, on-the-move interaction.

We demonstrate that such an interaction is more natural to the user than existing mobile interfaces.

Thesis Supervisor: Pattie Maes

Title: Associate Professor of Media Arts and Sciences

ReachMedia: On-the-move Interaction with Everyday Objects

by

Assaf Feldman

The following people served as readers for this thesis:

Thesis Reader____

Alex (Sandy) Pentland

Toshiba Professor of Media Arts and Sciences

MIT Media Laboratory

Thesis Reader ______________________________

Franklin Reynolds

Senior Research Manager

Nokia Research Center

Dedication

Masha my love, my passion and companion you are my best friend and I can't thank you enough for helping me and supporting me in this voyage and especially for suffering Boston for me, sorry.

Daniel and Mattan you are my little heroes and the motivation for this whole adventure.

Ima, Aba and Tamar. I owe you so much and I love you endlessly.

Acknowledgments

I would like to thank here the following people who I had the honor to learn from during these fascinating, challenging and rewarding two years.

First of all thanks to Pattie Maes whose kindness and wisdom have made my last year in the Lab a pure joy. Thank you Pattie for believing in me, and for guiding me in this project. I was honored to work with you and I'm grateful for getting to know you.

To Chris Schmandt who gave me the opportunity to join the Lab. Thank you Chris for all you thought me, and for being so considerate and empathic.

Special thanks to Sajid and Emmanuel I couldn't have done it without you.

To all my group members from both the "Ambient Intelligence" and "Speech Interfaces" groups, I learned from and was inspired by you and most importantly I enjoyed your company.

To Pat and Linda thanks for all your help and support and for enduring my incompetence, I'm sorry for all the troubles I caused.

To all media-labbers, students and faculty, thanks for challenging me and empowering me. You all form a culture, which I will admire and miss for the rest of my life.

To Nattalia Marmasse and the Zuckermans who were good friends and colleagues.

Table of Contents

Table of Figures

1 Introduction

With the growth of the Internet and the proliferation of digital information and services, many physical objects such as movies, books and other products now have extensive online information and powerful services associated with them, such as reviews, recommendations, and more. However, our interactions with this information almost always happen in front of a desktop computer using a mouse and a keyboard. We currently have no commonly available interfaces that allow us to combine the affordances and richness of our interaction with everyday objects, with the power and richness of the digital world.

Mobile devices such as smart phones have enough processing power and bandwidth to enable us to access these powerful services. However while this information may be accessible, it is not truly usable, mainly because the interaction modalities available in mobile phones are fairly limited. Current interfaces are based on the desktop WIMP (Windows, Icons, Menus and Pointing device) paradigm, designed for interfaces utilizing the user's full attention and hand-eye coordination, and not for interaction on the move. By "on the move interaction," we mean not just interaction that is hands and eyes free, but also requiring minimal mental effort. Interaction on the move by definition happens while the user is engaged in some primary task other than the interaction itself, such as walking or flipping through a book. Therefore such systems should be unobtrusive and usable everywhere, not just in the home or within some special infrastructure.

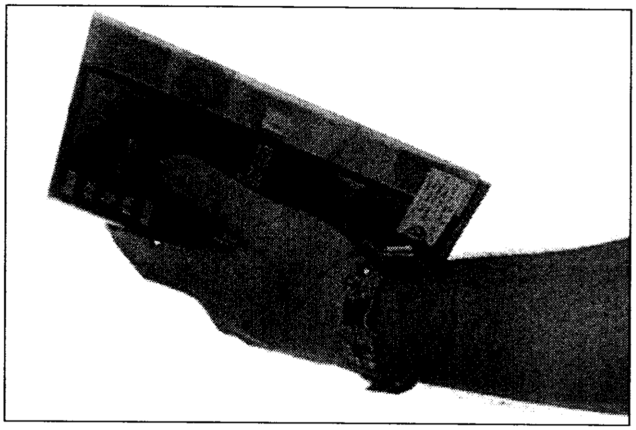


Figure 1: ReachMedia

Currently the only commercial hands-free and eyes-free solution available for mobile devices is speech recognition, which still suffers from poor performance in noisy conditions. But as noted by Brewster [Brewster 03], even if the performance of speech recognition technology were to improve, speech is not an ideal interface, because it is not socially acceptable. Not only do people feel embarrassed to use speech interfaces in public situations, speech is also intrusive to nearby individuals. This fact was indicated in a recent survey into the issue [Wasinger 05] and also by the growing phenomenon of cell-phone usage being banned in public locations such as buses and restaurants and the recent lack of enthusiasm from parts of the public to support cell phone use on planes.

Wearable interfaces do not usually fully support on-the-move interaction either. Jun Rekimoto [Rekimoto 01] defined two requirements for designing wearable devices that are going to be worn and used in everyday situations: first, they should support "hands-free" interaction in the sense that the device should not require the user to hold it in their hand, and even if the device requires the users to use their hands it should allow "quick changes between normal and operational mode". Second, they should be socially acceptable. Rekimoto remarks that wearable interfaces to date do not usually address these design requirements. They are often rather obtrusive as is the case for the twidller and glove-type input devices. Additionally, the problem of sensing context is often treated discretely, and many times depends on user input, which once again causes the system itself to become the focus of attention. Finally, the fine-grained manual dexterity that these devices require, demands that the user stand still to use the input device.

In our work, we are interested in interaction with physical objects such as books. As such, the interaction should not only be "hands-free" but also "eyes-free". Our goal for this project is to design a system that gives users seamless access to the information and services related to physical objects in a way that compliments the natural affordances of our everyday interaction with these objects. Recent work [Brewster 03] has shown the potential of gestural input and audio output for on the move interaction. In order to identify the subject of the interaction, we chose to use Radio Frequency Identifiers (RFIDs). RFID tags are passive chips with a unique identifier that are powered by inductive power transfer and capable of storing, receiving and transmitting data. They are relatively cheap, unobtrusive, and do not require line of sight for reading.

We present in this thesis a system, that puts together the concepts described above, namely, subtle gesture recognition interfaces and a body worn RFID. The center of the system is a wristband with an RFID reader that can detect the object that is held by the user. The system can deliver information about this object and allows the user to interact with this information through the wristband using a gesture recognition algorithm.

1.1 Contribution

The contributions of this thesis include a body worn, wireless RFID, subtle gesture recognition interface and some initial testing of the concept. .

2 Scenarios

Tamara is entering a bookstore. She heads to the section she is interested in, and finds the book her friend had recommended. As she holds it, the wristband's LED lights up, indicating that it has identified the book. A few seconds later, as Tamara is already midway through reading the description on the back of the book, a slight wind sound, played to her through her tiny fashionable Bluetooth earbud, informs her that the system has information ready for her. By the pitch and volume of the wind sound, she knows that the system had ranked the information as highly relevant.

After she finishes reading the back-cover she flicks her wrist downward (corresponding to the "select" signal), and the LED on the wristband lights up indicating that the gesture was recognized. Half a second later the voice menu says "reviews", she signals "select" again and the voice says "New-York times book review" (her favorite source of book review). A slight inward twist with her wrist signals "next", and the voice says "Wall street journal". She flicks "select" again and the review is played back to her as she is flipping through the book. When the playback is over she twists her wrist slightly outward signaling "previous" and then listens to the New-York Times review. By now she had already finished skimming through the introduction of the book and checked the table of contents. She decides she wants the book but the price is a bit too high, she takes out her phone; the screen shows a list of information services relevant to the book. She clicks on the "other formats" option and sees that the book is also available as an e-book and a paperback. She checks the paperback's price and finds out that it is in her range, however it is not available in the store at the moment, so she orders it to arrive by mail.

When Tamara gets the package she scans the RFID that is on it and by doing so enters the book in her smart-home library. This allows her to make virtual bookmarks in the book and leave a message in the book for her friend who requested to read it after her.

3 System Overview

This thesis presents a novel system built around a wireless wristband that includes an RFID reader, 3-axis accelerometers and RF communication facilities for data transmission. We view the wristband as a part of an on-body network that includes a personal network-connected device, e.g. a cell phone and a wireless earpiece. The wristband allows the user an implicit, touch-based interaction with services related to objects by using RFID.

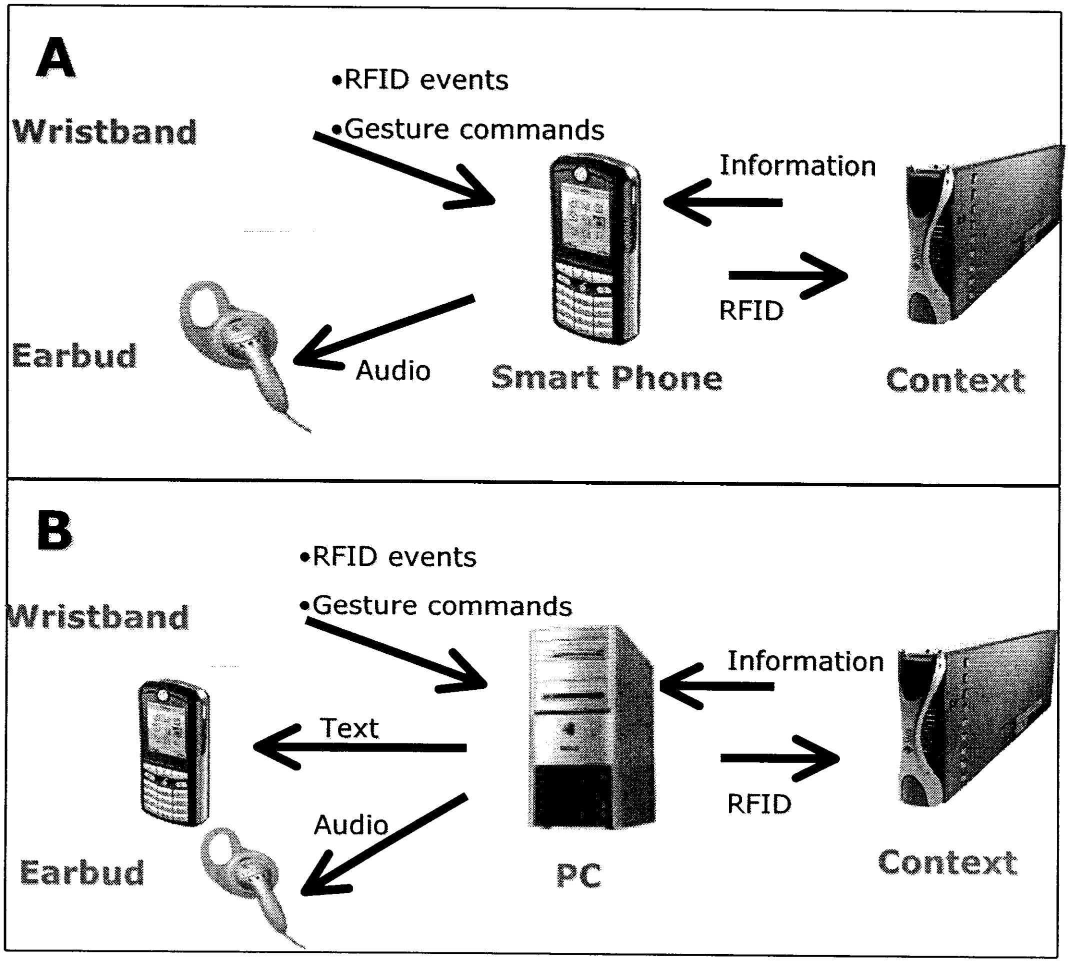


Figure 2: Envisioned architecture (A) versus the current implementation (B)

Once an object is detected in the user's hand, the user can interact with the system using slight wrist gestures, i.e. continuous movements of the hand in space. Thus, allowing an unobtrusive and socially acceptable interaction with everyday objects, yet a rich and engaging one. As mentioned above, we envision ReachMedia as a system that is mediated and coordinated by a personal computing device such as a smart phone (Figure 2: Envisioned architecture (A) versus the current implementation (B)), however, because of time constrains the implementation of ReachMedia is currently done on a PC, with the smart phone acting just as an output device, presenting text to the user. Nonetheless, the design and architecture of the software took into consideration the end goal of running the system on a low power processor and used techniques that are suitable for such processors, with the intent of making the software as portable as possible to these less powerful devices.

4 Related work

There exists a rich body of work, which uses sound, head movement, body gesture, and augmented/tagged objects to trigger and manipulate interaction. Early work is the field used cameras, color tags and visual displays to create an augmented interactive reality.

Such designs include Nagao's [Pirhonen 02] speech-based "Ubiquitous Talker," and Starner's [Starner 97] "Physically based hypertext," where stepping toward an object indicates hyperlink activation.

A very recent one - called ShopAssist [Wasinger 05] addresses the problem of augmenting the physical activity of shopping in a "electronics boutique" and focuses on examining few input modalities. It uses an RFID reader embedded in a shelf and a PDA with 2d gesture recognition, GUI and speech interface to allow users to interact with services and information related to RFID tagged products in the store. However, ShopAssist does not use mobile RFID input or gestures the way we defined them. Nonetheless a very applicable result of that project to our system is their finding that although speech was a preferred modality in laboratory tests it was found, in their test field, to be unsuitable for public scenarios.

We will review other related work separately while focusing on the individual technologies that together make up our system.

4.1 Multimodal interfaces for on-the-move interaction

The widespread usage of mobile phones and the realization of the limits of their input and output modes, i.e. their small screens and keyboards, highlights the need for new interaction methods for mobile devices. This realization has given rise to extensive research in the field of multimodal interfaces to enhance the usability of mobile devices and to enable a new level of services and information for users that are on the move.

Some of the key research that our work builds on is Brewster's investigation of the combination of gesture input and audio output for interactions on the move [Brewster 03]. In this work two systems were presented and evaluated; both used speech and non-speech audio output in combination with gesture input. The first system was a 3D audio radial pie menu that used head gestures for selecting items. The second system was a sonically enhanced 2D gesture recognition system for use with a belt-mounted PDA. Their results show that sound and gesture can significantly improve the usability of a wearable device in particular under 'eyes-free', mobile conditions.

The use of voice and audio interface for mobile systems is not new. Sawhney and Schmandt [Sawhney 00] previously presented a 3D spatialized audio system, "Nomadic Radio," which used voice and non-voice input and output methods. The usage of gesture in a wearable context was also explored and discussed in Rekimoto's GestureWrist [Rekimoto 01]. In this work, a wristband with capacitive sensors was used to recognize hand gestures and forearm movements, focusing in particular on minimally obtrusive interaction. Preliminary work investigating subtle and intimate muscle movements and EMG has also shown great potential for on-the-move interaction [Costanza 05]. However, most of the work in gesture recognition relies on interfaces that we consider unsuitable for the types of applications we are interested in. Many systems use a glove interface, such as in [Baudel 93] which in our mind is too constrictive an interface for usage in everyday life.

In recent years accelerometers have been gaining popularity as mobile sensors due to their low price, low power consumption and robustness. However, comparatively little work has been done applying accelerometers to the gesture recognition problem. In particular, Benbasat [Benbasat 01] has created a framework for rapid development of discrete gesture recognition applications. Also, Mäntyjärvi [Mäntyjärvi 04] has used mobile phones, enhanced with accelerometers, to allow the use of gesture control, providing some highly relevant insights into this topic. Hinckley [Hinckley 05] has used static 2D acceleration signals (tilt sensors) to allow background sensing of user actions and in turn to allow a device to take proactive action. Lastly, body worn accelerometers were also used in research on activity detection and recognition [Lukowicz 04].

4.2 RFID augmented objects

Using RFID for mobile interaction was suggested by Hull [Hull 97] for tagging places and people, Want [Want 99] further explored the technology and suggested using it to enable interaction with physical objects, which has since become a popular line of research [Wan 99, Florkemeier 04] in the pervasive computing community.

Typically, mobile pervasive systems that use RFID for augmenting everyday objects have relied on explicit (i.e. intentional) actions of the user. Often this design decision has been forced by the fact that the input and output devices used in these projects were PDAs that were too big to support implicit input. However, recent work at Intel Research Labs has looked into activity detection for health care application [Philipose 03] using an RFID reader integrated into a glove.

5 Interaction design

One of the first design decisions we made was about the form factor of the device. We wanted the device to be simple and elegant, an accessory that we could imagine people wearing in public in everyday situations. Although a glove is a very effective form factor for both RFID [Philipose 03] and gesture recognition [Baudel 93], we believe it to be too restrictive and inappropriate for everyday usage.

Advances in technology have significantly reduced the size and power consumption of all the components mentioned above, making it possible to consider small and elegant wearable devices. The wristband approach seemed more realistic especially because wrist-worn accessories, such as wrist-wallets and wristwatches are already a part of our common apparel. Given the current trend in miniaturization of electronic components, it is also reasonable to assume that the hardware would easily fit inside one of the above accessories within a few years.

In the next sections we will now detail our design considerations for the two major aspects of the interaction, namely the user interaction with the objects themselves and the user interaction with the information and services.

5.1 Touch based interaction with objects

In designing the interaction with the system, our primary challenge was to ensure that the affordances of the natural interaction with objects would be minimally affected as a result of the addition of the digital dimension. We therefore chose to base the interaction on the action of touching or holding an object, which also allowed us to make the detection process implicit, thus removing the intrusive process of explicitly having to 'scan' objects for interaction.

In choosing RFID as the tagging technology, we saw two main advantages for on-the-move interaction. First, unlike barcodes or infrared beacons, which require line of sight and explicit scanning, RFID tags are read using radio waves, thus allowing implicit event triggering. Users can concentrate on their primary task while the RFID reader detects the tag on the object they are holding or touching.

Second, the cost of the infrastructure required to support an RFID based system is low. RFID tags are battery-less and cheap. It is also expected that RFID tags will in the near future be used widely in retail, particularly for mid-range products such as books, because of their advantages over bar codes. Therefore in many cases the basic infrastructure incurs no extra cost.

Unfortunately, RFID technology also has its limitations. Primary among them is the reading range. We discuss the engineering aspects of this problem at length in the hardware section, but the interaction design aspects also merit discussion. Because the system is designed so that the detection of objects is implicit, the range of the reader is of great importance. If the range is too large, many events will be triggered, and many of them will be "false positives." For us, the form factor of the device has necessitated investigation into improving range, because if it is too short, the user will have to essentially scan the objects explicitly, and the detection process will require more attention than we consider desirable.

One of the primary design goals of this project is to provide an unobtrusive experience. Therefore, we chose to err on the side of caution in choosing a range which would limit the total number of "false positive" interruptions. Thus the interaction is "semi-implicit" meaning that we assume some type of conventions in the positioning of tags. On books for example tags might always be placed on the bottom of the spine and users will have to be aware of that if they with to use the system. Such a convention will allow the tag to be implicitly read when a user holds a typical book in a natural way. Some very large books may require the user to pay attention to how they handle the book to get the desired effect.

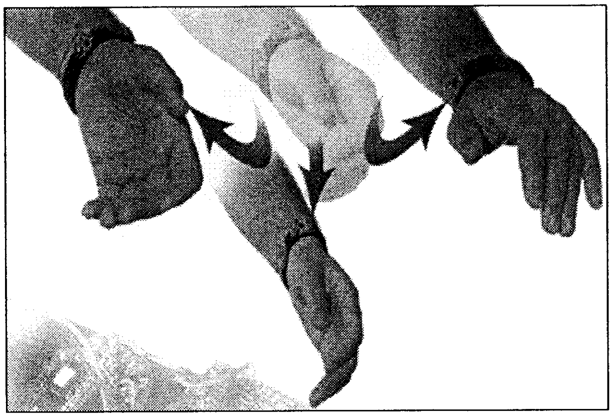


Figure 3: The three gestures – Clockwise rotation, Counter clockwise rotation and a downward flick

5.2 Gestural interaction with information

In choosing the mode whereby the user interacts with the system, a number of technical and human factors came into play. The final choice of input modality was motivated by social acceptability, intuitiveness, and separability. Although speech is a totally hands-free input modality i.e. doesn't require any usage of hands, as mentioned previously, we considered speech not to be ideal for input for our scenario since it has been found both by observation and research to be socially unacceptable in many public situations [Wasinger 05].

We chose to use gestures for this project because they can be made quite minimal and thus socially acceptable, while retaining ease of use. In choosing gestures for the system, we focused on gestures with as little as possible human-to-human discursive meaning. Head nods, for example, when performed in a public space might be misinterpreted by nearby individual. As such, we chose to focus on using minor wrist motions, which are outside the normal focus of attention.

In considering sensors for this project, we found EMG, although extremely subtle, is still not robust enough and also somewhat invasive. We therefore chose to use accelerometers as input sensors for the wristband. We found rotations of the wrist to be minimal in terms of the amount of movement in space, yet highly recognizable by inertial sensors.

We chose a minimal set of commands, which we regarded as sufficient for supporting navigation of simple menus. The commands we chose to represent with gestures are: "Next," "Previous" and "Select." The gestural metaphors we used for these commands are slight flicks of the wrist of the non-dominant hand, so chosen such that the user can continue to hold the object while manipulating it and interacting with the menus, though either hand may be used in practice. As an example, inward rotation of the wrist is "Next", outward rotation of the wrist is "Previous" and a downward flick is "Select". "Select" on a node causes the user to move down in the hierarchy. Moving up in the hierarchy is achieved by using the "previous" command when positioned on the first element in the hierarchy. Using "next" on the last element in the hierarchy causes the menu to wrap-around and move back to the first element in the hierarchy.

Our voice menu is a simplified version of the common Interactive Voice Response (IVR) type of menus, which use speech labels and cues to guide the user through the hierarchy. Unlike IVR, the system does not read the list of options when entering a new level in the hierarchy. Instead, each node just states the number of option it includes and the user can iteratively move inside the hierarchy using the "next" and "previous" commands. The leaves of the hierarchy are the results of the various services available for the object the user is interacting with. The context manager provides text to speech output for all nodes and leaves directly from the output of the services. Following the result of Pirhonen [Pirhonen 02] we added non-voiced audio feedback to indicate the type of gesture detected as well as when an object is detected. Non-voice cues are also produced when the first or last items in the current hierarchy are reached.

An interaction with a book, for example, will take the following form: the user picks up the book, a slight notification sound indicates to her that the system has found services related to this book. The user flicks her wrist downwards, the short non-voice "select" cue is played and immediately after a voice prompt says "ratings", she flicks her wrist outward and hears the "next" command signal and the voice prompt says "reviews"; she signals "select" and the voice prompt says "New York Times", she signals "select" again and listens to the review while flipping through the book's pages.

6 Hardware

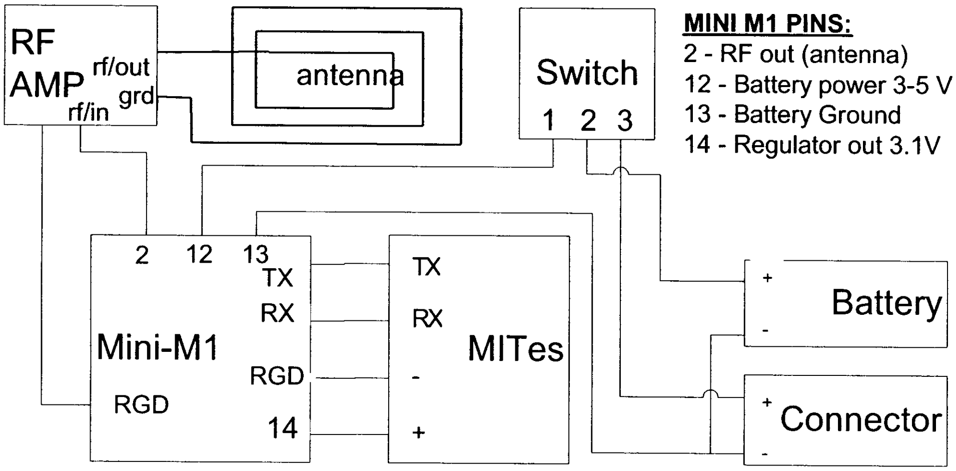


Figure 4 Hardwaere scheme

The hardware for the wristband consists of an off-the-self RFID reader. It is managed by a customize version of the MITes board from MIT's House_n, which also includes the RF transmitter with accelerometers.

6.1 RFID Technology

The RFID reader used is a SkyeTek M1 Mini, which has a diameter < 25mm and thickness < 2mm, which is optimal for the project requirements. The reader includes an integral 3V regulator, which powers the MITes, and is in turn powered by a 4.7V prismatic lithium polymer rechargeable battery with off-board charger for long-term use. The MITes microcontroller controls the reader via a serial connection using the SkyeTek m1-mini ASCII based protocol.

While RFID tags provide an excellent source of context information, they also pose problems. The tags used in this project are ISO15693 compliant 13.56MHz chip tags using inductive power transfer. Their availability and price are optimal for ubiquitous applications.

However, they experience problems when bent (i.e. when applied to the spine of a thin book).

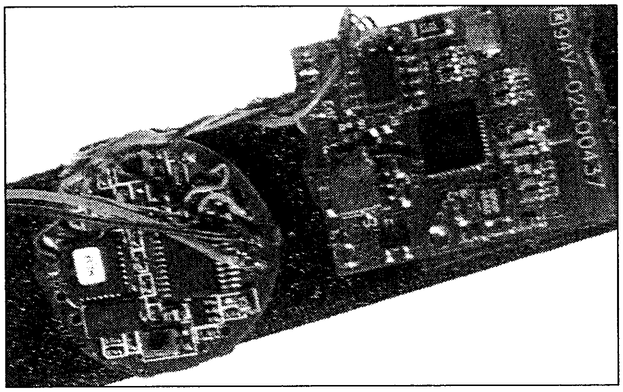


Figure 5: ReachBand includes the MITes board (right) and the Skyetek m1-mini RFID reader (left)

Similar industrial and academic systems, like Wan's [Wan 99] *'Magic Medicine Cabinet'* or smart bookshelves, use RFID readers that are embedded into static objects. This makes it possible to use fixed reader antennas, which usually does not exhibit these pathologies due to power availability and the more extensive drive electronics.

Unfortunately, the 2.3cm diameter of the integral antenna, combined with low drive current for the outgoing carrier causes the effective range of the standalone reader to be as small as 4 to 5cm. To increase the range, the wristband form factor was used to increase the antenna diameter to the diameter of the wrist. However, the reader by itself failed to produce sufficient output to drive this antenna. To that end, an RF-mode amplifier was designed to increase the output of the reader. This has doubled the range to approximately 10 cm. While still 3cm shorter than our goal of 13cm, it is sufficient to allow natural interaction with objects such as books. We are currently investigating methods of separating the input and output waveforms and bi-directional amplification methods, in order to further increase the range.

6.2 Antenna Design

The unique wristband form-factor that we use in this system raised a few challenges when building the antenna specifically because the antenna uses four loops of coil, therefore the mechanism that is used to close the wristband around the wrist had to also close the circuits and connect the wires into a continuous coil that will loop around the wrist.

Our solution was to use snap-on metal buttons, to which the coils were soldered. Four buttons were connected to each end of the wristband (one side with mails and the other with females). The buttons are connected between themselves with coils in a way that ensures that once the buttons are snapped together a continuous four loops coil is formed around the wristband.

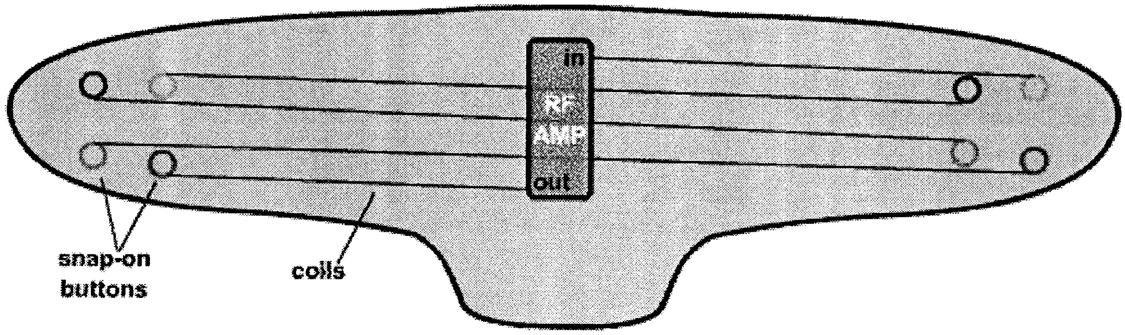


Figure 6: Antenna design

6.3 MITes

For the RF and accelerometers this project uses the MITes wireless sensor [Munguia 04]. The MITes wireless sensors are designed around the nRF24E1 chip manufactured by Nordic VLSI Semiconductors. The nRF24E1 integrates a transceiver, an 8051 based microcontroller running at 16Mhz, a 9 input 12-bit analog to digital converter, and miscellaneous peripherals (3 timers, UART, SPI, PWM, and 11 IO pins). The transceiver operates in the 2.4GHz ISM band (available worldwide without a special license), offers data rates up to 1Mbps, and provides 125 Tx/Rx channels for multi-channel communication. It also offers low power consumption operations such as shock burst and sleep mode and low Tx/Rx power consumptions of 10.5mA at -5dBm and 18mA at 250 Kb/s respectively. Finally, its cost is $6 per unit or $3 in quantities of 10,000 units. The chip uses a proprietary protocol that does not interfere with 802.11 or Bluetooth, which for the MITes has been confirmed in practice. The MITes also include an external 4K EEPROM program memory, ADXL202/210 accelerometers and a 50 ohm antenna matching circuit between the nRF24E1 RF I/O and the onboard microstrip antenna.

Wearable MITes – Mobile MITes are MITes with ±2g accelerometer, and an extra side daughter board to provide the third axis of acceleration. They can be placed on different parts of the body to measure acceleration. To the best of our knowledge, mobile MITes are the smallest, lightest, and least expensive wireless 3-axis accelerometer sensors available to the research community. Their dimension (1.2 x 1.0 x 0.25in), and weight (8.1g including battery) permit them to be embedded or attached to wearable objects, such as watches, shoes or belts, without constraining the wearer's movements.

The cost of a single prototype is currently only $41.20. This enables a day of continuous data collection without replacing batteries. The sampling rate could be decreased to extend battery life if required. Currently, it is possible to receive acceleration data from six mobile MITes simultaneously, each transmitting on a different channel. However, it is be possible to extend this to up to 125 mobile MITes simultaneously.

To achieve low power consumption and extended battery life, the microcontroller and associated circuitry are kept in sleep mode whenever possible. When the acceleration signal needs to be sampled, the microcontroller wakes up, turns on the accelerometers, reads a sample, transmits the data, turns off the accelerometer, and returns to sleep mode.

Wearable RFID Mites – the RFID MITes are an aggregation of the wearable Mites and include an addition code for driving the Skyetek RFID reader. We use the ASCII mode of the Skyetek Mini-M1 protocol version 2. The MITes processor polls on the RFID reader at a rate of 5hrz requesting it to perform a SELECT_TAG operation. Responses come back through the serial connection between the MITes and the reader, and the MITes processor parses them and extracts the Tag ID. The ID is then sent to the receiver MITes via the Nordic RF. The ID that is sent includes only the last (most significant) 30 bits out of the full 48 bits (6 Bytes) of the RFID tag. The reason we send only partial data is due to the limits of the MITes packet size and communication protocol, however for our prototyping needs 30 bits are sufficient.

Receiver MITes - The MITes receiver interfaces with the RS232 serial port of any PC, laptop, or PDA. It includes a RF24E1 MCU+transceiver, a RS232 level converter (MAX3218) and a voltage regulator (MAX8880) for external power supplies between +3.5 and +12v.

7 Software

ReachMedia is composed of 2 main software components: the MITes driver, and the client. The MITes driver processes the data coming from the wristband; the client renders the information and drives the interaction with the user. These two components are mediated by the ContextManager (Sadi 05), which acts as a control point, it both establishes a session between the MITes driver and a client, and resolves the RFID events for static objects.

7.1 MITes Driver

The MITes signal processor and classifier software component of ReachMedia is in the core the system and is a prototype for the software that is supposed to be running on the smart phone. Its main tasks are to execute the gesture recognition related tasks and to communicate with the other distributed components. Its direct functionalities includes:

- Process and classify the acceleration signal.
- Allow a UI for training and testing the gesture classification process
- Provide a visualization tool for the gesture classification.
- Communicate with the wrist band
- Communicate with the ContextManager.

7.2 Gesture Recognitions - Real-Time Processing and Classification

The gesture recognition system was implemented in Java mainly in order to ease the integration of ReachMedia with the pre-existing MITes project. However, there are few advantages to the usage of Java regardless of integration concerns, mainly due to the elegant sockets, serial and multithreading APIs.

In the next paragraph we will review the components of the system and will discuss the architecture design and implementation concerns.

The gesture recognition system is a real-time system. By real-time we mean that the system is composed of a collection of periodic activities that cooperatively access shared resources, together with a set of activities that respond to external or internal events. Essential to this purpose is a thread/task model for expressing concurrent activities; mutual exclusion for shared resources, inter-thread or task coordination, and responses to asynchronous events including hardware interrupts.

Although the standard Java SDK has a good enough multithreading model, at least for prototyping needs, it is not considered a perfect choice for developing real-time systems mainly because of its garbage collection mechanism that freezes all the processing in the VM in an unpredictable way. This freezing makes Java an unsuitable development tool for strict Real-Time systems that require very precise scheduling such as medical system. Our system however is not very strict and can cope with a slight loss of data. Therefore we decided to stick to Java for the gesture recognition system, and designed it in a generic and modular way, with the hope that this part of the project will be reusable for other systems that require real-time signal processing and classification.

The system is composed of four main classes, which represent the algorithmic logic of the system

Feeder - is in charge of the communication with the external hardware device (the MITes receiver) through the serial port.

Activity Recognizer- is in charge of getting the signal samples form the feeder and segmenting the signal into activities i.e. gestures.

Feature extractor - is in charge of taking the gesture segments from the activity recognizer and extracting the feature vector out of it.

Classifier – is in charge of taking the feature vector from the feature extractor and using it to do the actual classification.

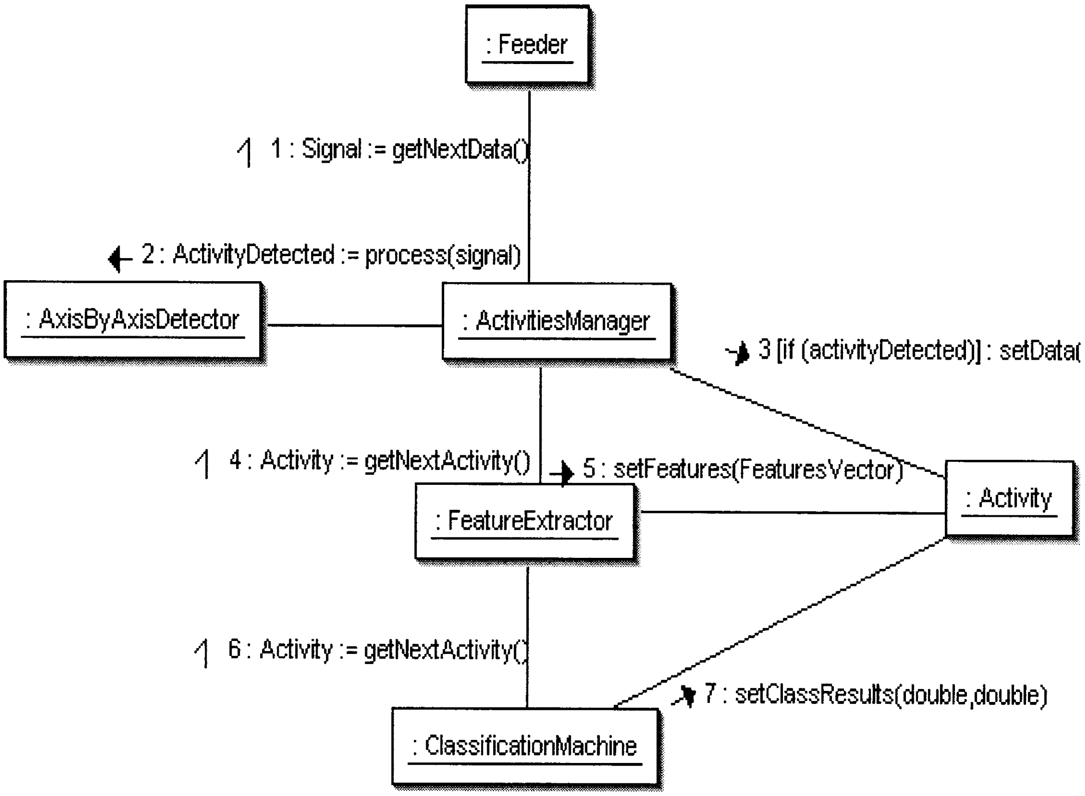

: Feeder
1 : Signal := getNextData()
2 : ActivityDetected := process(signal)
: AxisByAxisDetector
: ActivitiesManager
3 [if (activityDetected)] : setData(
4 : Activity := getNextActivity()
5 : setFeatures(FeaturesVector)
: FeatureExtractor
: Activity
6 : Activity := getNextActivity()
7 : setClassResults(double,double)
: ClassificationMachine

Figure 7 Data flow in the gesture recognition system

As can be seen in figure 11 the activity class, which stores all the information related to a gesture activity is "gluing" these components together. The activity class is passed from component to component, each one stores the result of its processing in the object and passes it to the next component which then uses that result to do some further processing (see Appendix A – data flow in gesture recognition system).

Because each of the above components relies on the result of it's successor, and at the same time is suppose to work in a concurrent fashion, the system has to be asynchronous (non blocking), so that the classification intensive processing will not block the system from receiving new data. Each component therefore has to run in a separate thread, which raises the usual challenges of threads synchronization and monitoring.

Mutual exclusion for shared resources and inter-thread coordination, is achieved by using Java's lock and monitors so that the threads do not need to poll on one another while waiting for new data to come in. This type of asynchronous event handling is more efficient and is essential when designing software for low power processors. This need is emphasized in ARM processors that have varying clock speed, because polling can prevent the processor from reducing its speed.

The system uses a variation of the "consumer provider" pattern to achieve synchronazation (see Figure 8: UML sequence diagram for the producer-consumer pattern implementation). The Feeder, Activity Manager and FeatureExtarctor classes are what we call "providers" and the Activity Manager, FeatureExtarctor and Classifier are what we call "consumers". A provider is a component that queues data for a consumer (in the current implementation, we are assuming a one-to-one relationship between consumer and provider). So in our case for example, the consumer of the Feeder is the ActivityManager, which is also a provider himself, as he is providing input to the FeatureExtractor.

The providers all have a member variable from the SynchedObj class. The SynchedObj class is a linked list that queues the data that the provider produces in a FIFO (First In First Out) manner.

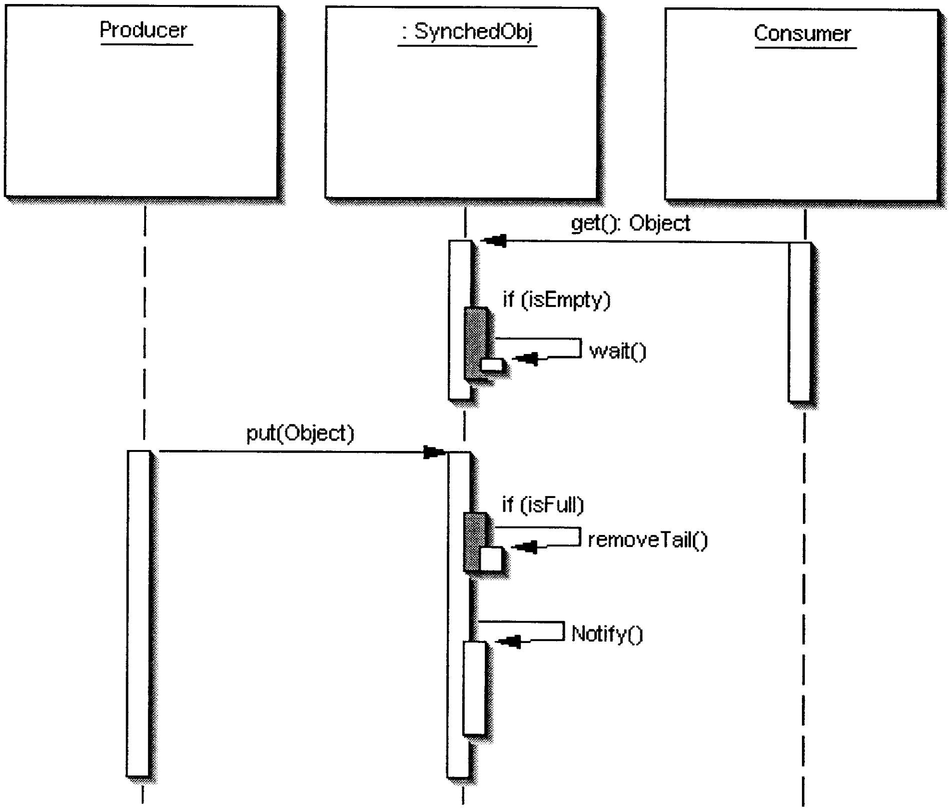


Figure 8: UML sequence diagram for the producer-consumer pattern implementation

The SynchedObj class also has a maximum capacity for this queue, so if the capacity is reached new data will replace the old data and old data will be lost. When a consumer thread is requesting data through the GetNextData() method, it will get the oldest item in the queue (the tail of the list). If the queue is empty the consumer will just get blocked until new data will come in. For the thread blocking we use Java's wait() function.

Once the provider inserts new data into the queue it calls Java's notify() that wakes up the locked consumer.

7.3 RFID event handling

RFID events are received by the Java application from the receiver MITes, the application then parses the events and extract the reader and tag IDs out of them. The reader's ID is taken from the ID of the RF channel on which the wearable MITes sends the RFID event..

The tag ID and reader ID are then sent over TCP/IP to the ContextManager, which resolves it and notifies the appropriate client.

7.4 Context Resolution Service

Mobile context-aware devices pose an interesting set of software design choices. Due to the limitations of the mobile interaction platform, it is usually necessary to handle the management and processing of complex or resource-intensive context information off-board, and the need for "context services" for such applications has been long recognized [Abowd 97, Cohen 01]. While there has been some dissent from those interested in automation, in general, an off-board processing model provides many benefits in terms of mobile node and communication protocol complexity, and is the model used in the Context Manager, the object resolution and service management framework designed by Sajid Sadi as part of his research in the group and used in the ReachMedia project.

The Context Manager has a threefold task of converting RFID identifiers into object metadata and context, accessing services with the available context and session information, and transferring the context information back to the device for display. This is done with "connectors" that translate the incoming RFID sensor data into internal session data, and then converting the output of the context-relevant services for the output modality. By separating out object resolution, it is possible to minimize wristband data bandwidth, while reaching a compromise that allows many services to use the data [Chen 02]. This has the additional effect of reducing the demands on the RFID tags, which can now contain minimal metadata. On the output side, the output connector capitalizes local processing to provide optimally translated output, thus reducing overhead and requirements for the target mobile device.

7.5 Client

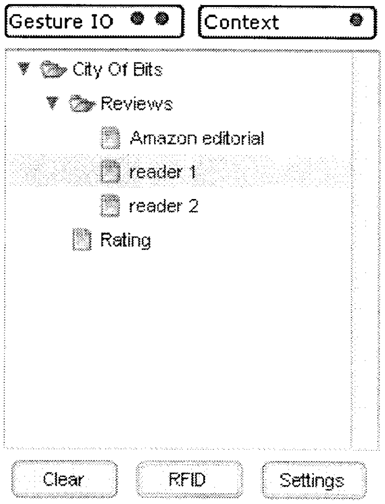


Figure 9: Flash Client menu visualization screen shot

The client is the software component that parses and renders the information and services that are related to the physical objects; it listens for events coming from the "Context Manager", and it holds the menu's state.

When initiated the client registers itself with the "Context Manager" and waits for RFID events to arrive via TCP/IP. The RFID event includes an XML file that lists the services and information related to the physical object. From that point on the "Context Manager" routs to the client all the user commands coming from the wristband.

The implementation of the interface and the menu navigation system (as described in the "Gestural interaction with information") was done using Macromedia Flash.

7.6 MITes Driver GUI

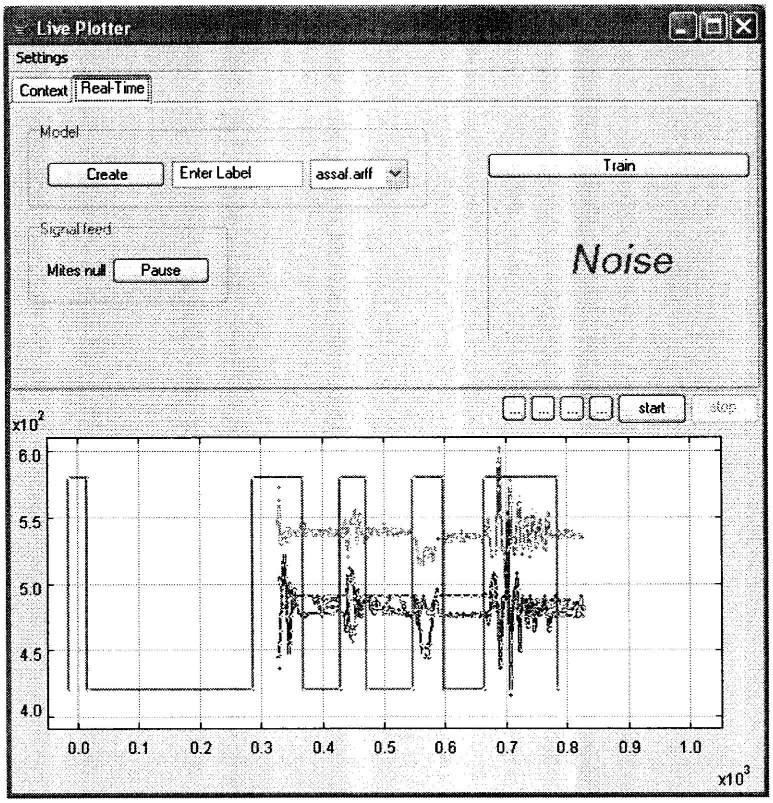


Figure 10 : The GUI screen with the Real-Time panel tab chosen, and the LivePlotter showing both the signal and the activity detection boundaries (in orange).

The GUI was designed and used mainly for debugging and running the evaluation of the gesture recognition system, it includes an interface for setting the properties of the system, an interface for training new models, and a visualization of the acceleration signal processing and classification output.

The interface is composed of two main components a tabbed panel and a plotter grid, the tabbed panel has two tabs named "Real-Time" and "Context".

7.6.1 Live Plotter

The plotter plots the real-time input signal and the activity detection edge boundaries. The plotter uses the PTOLEMY library and utilizes its real-time XOR based optimised PlotLive class. The plotter plots the 3-axis signal continuously and also shows the gesture boundaries that are detected by the activity detector. (See Figure 10 : The GUI screen with the Real-Time panel tab chosen).

7.6.2 Real-Time Panel

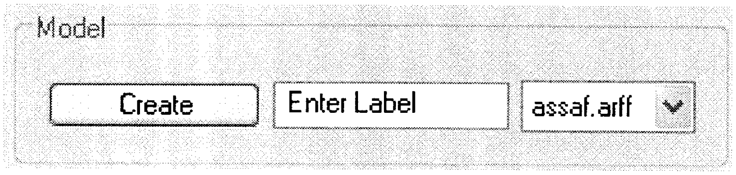


Figure 11 : Model panel

The real time panel include all the interfaces related to the gesture recognition system, it allows the user to train a model, set a pre-trained model, view classifications and label gestures.

The interface is divided into few areas and groups: on the upper left side there is the Model" inner panel, bellow it is the signal inner panel and on the right hand side there is the "train" button and string output display area.

The model inner panel

The model inner panel includes few controls that allow the suer to set and create the model of the gesture recognition system. It includes the following fields:

Model selection box – a selection box that displays the current model used by the system for gesture recognition and also allows the user to choose another model from a list of all the currently available models. The box is populated when the application starts, by looking into the "model's directory", and finding all the files whose names end with ".model". If a "model's directory" is not found at start time at the default relative location, the system will ask the user to choose a directory from the disk. When a user created a new model while using the application the new model is automatically added to the list and selected as the current model for the classifier.

Label text field – The label text field allows the user to manually label a gesture at real-time. The system will use the content of the "label" text filed as the label for the detected gesture. The field is updated by the UI when ever a the 0-4 key strokes are captured, each key stroke is mapped to a gesture in the following way:

0- No label.

1- Next.

2- Previous.

3- Select.

4- Noise.

Create Button – The create button saves all the current gesture training samples and their related data and trains a model based on these samples. When the button is pressed the system prompts the user to select a name for the newly created filed. As a result of this action few files are created. Every file name will be composed from the name the user entered than a dot followed by the classifiers file type string and finally the file type string., i.e. following structure <user chosen name>.<classifier string>.<file type>. The classifier strings are "arff" for the weak Naïve Bayes and "dat" for the LIBSVM one.

The file types that are created are:

- *.classes file* – this file contains the classification produced by the current classifier, this file is important when testing a model because it contains the current model prediction for the gesture, which can then be compared to the true labelling of the gesture.

- *Features files* – this file has no extension and it contains all the feature vectors formatted in a manner specific to the current classifier
- *.mat file* – this file contains all the feature vectors in a classifier agnostic coma separated style.
- *.model file* - this file contains the model in a format specific to the current classifier.

7.6.3 Context Panel

The context panel is an interface for setting the properties of the system related to the connection of the system to the Context Manager. It includes the following fields:

IP text field: sets the IP used by the system to connect to the Context Manager server.

Port: **text field**: sets the port used by the system to connect to the Context Manager server.

8 Evaluation

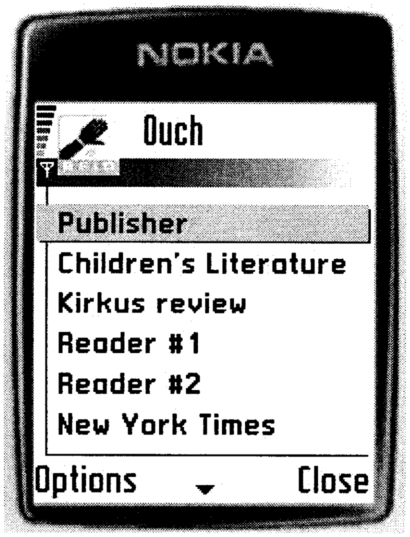
In this chapter we present the results and observations gained from a user study that was conducted in NY on August 5th and 6th 2005. We conducted an experiment to attempt to quantify the advantages and disadvantages of the ReachMedia interaction design and implementation, compared to a standard mobile interaction through a commercially available smart phone. Because ReachMedia provides hands-and-eyes free interaction we assumed that it would have some advantages when attempting to keep users "in the flow" of their interaction with a physical object.

Flow, a concept coined by Csikszentmihalyi [Csikszentmihalyi, 91], is in essence a state of deep cognitive engagement people achieve when performing an activity that demands a certain level of focus, like writing. Measuring how well an interface keeps users "in the flow" is hard, especially in the case of the relatively short everyday interaction that we have addressed in this project. Bederson [Bederson 04] suggested measuring the flow of an interface using Czerwinski's relative subjective duration (RSD) [Czerwinski 01], a percentage difference between the actual and perceived task duration, i.e. time to completion of a task. Czerwinski observed that as a task becomes more difficult (perhaps due to an interface design), participants will likely overestimate how long that task took. In contrast, if participants can complete the task easily, they are more likely to underestimate how long that task took in comparison to the actual task time. In addition, participants do not necessarily know ahead of time which direction the experimenter expects the time estimates to go, and hence may not "bias" their reported estimates toward the positive end of the scale, as so often happens during lab studies using satisfaction measures in questionnaires.

We also measured cognitive load explicitly by using the NASA TLX subjective workload assessment questionnaire, which was embraced by Brewster [Pirhonen 2000] and is gaining popularity in the pervasive computing community. (See Appendix B – NASA TLX).

8.1 Design

We used 11 subjects, 7 males and 4 females ages 24-35, with no pre-knowledge of the project. Each session lasted between 30-40 minutes. We used a within-subjects design, with each subject performing the task under two conditions. Our independent variable was the device type. We analyzed the results per measure with a one-factor ANOVA. We chose to compare ReachMedia to a smart phone, being the closest commercially available on-the-move device. We used the phone as a base line for the RSD variable and for the NASA TLX measure.We assumed that the phone would be less cognitively demanding to operate, because of its visual interface and given the user's relative familiarity with the device and its conventions. However we hypothesized that ReachMedia would have some advantages in this task such as allowing the user to flip through the book while using the device because of its hands free capabilities.



Phone condition– We created an application running on a commercially available phone, with similar functionality to ReachMedia's menus. The application was not able to detect objects, thus, choosing the book was done manually by clicking on a predefined menu. Once a book was chosen the application allowed the subjects to listen to audio reviews of the book, by clicking on items from a list (see Figure 12: Phone condition menu).

Figure 12: Phone condition menu

A click triggered a playback of the review's audio. The application was written in java (J2ME MIDP 2.0) and was running on a smart phone (Nokia 6600) using the standard Series 60 menus.

ReachMedia condition– We used ReachMedia as described in the interaction design section. The actual menu was however very simple and consisted of one folder with reviews.

8.2 Task

For this experiment we constructed a simple task that users could learn and perform easily in the timeframe of the experiment. The task required the subjects to flip through a book while interfacing with a device that provides audio reviews about the book. The experiment included two conditions for each condition a different device was used for interfacing with the audio menu.

From the point of view of the subjects we wanted the task to be engaging and centered around the physical object rather than the interaction with the devices. We tried achieving that by a fictional scenario, in which the subjects were told that they were invited to a birthday party of their friend's kid, and they needed to buy a book as a present. Their assignment was to choose one of two books, given a pretty abstract description of the parents and the kid. They were asked to examine one book at a time and assess how appropriate it would be as a present, while interacting with both the physical book and with an interface to audio reviews about the book. Subjects were told that they would be scored based on their choice of book, and the sum of the durations of their interactions with each one of the books (the shorter the time the higher the score).

It is important to note that from the point of view of the experiment there was no preconceived notion of which book is fit for the kid/family. Additionally, we purposely gave a description of the kid/family that had little to do with either of the books, or was equally applicable for both of them, in an attempt to keep the subjects looking for information in and about the book.

In the choice of children books for the tasks we aimed to encourage the subjects to engage in flipping through the book. To achieve that we chose illustrated and rather long children books with about 40 pages each, so that the books were too long to be read from beginning to end in the experiment's timeframe, yet short enough for the subjects to understand the basic narrative and style by flipping through them and looking at the illustrations.

The emphasis of the scoring by time was meant to give the subjects an incentive to simultaneously use the device and flip through the book. We did record the time though for the RSD measure.

We required the subject to standup and hold the book in their hands in an attempt to create conditions as close as possible to a bookstore type of setting.

8.3 Materials

All audios were created using AT&T Natural Voices Text To Speech application. All the texts were taken from Amazon's book reviews, and consisted of the publisher's description, two newspaper reviews and two reader reviews.

8.4 Procedure

The evaluation started with a short introduction of ReachMedia to the subjects. We then asked the subjects to train a model using the gesture recognition system training software. Next, subjects were introduced to an example menu, and got a couple of minutes to practice using ReachMedia until they felt confident about their ability to use it.

We then moved to the actual tasks. For each task the subjects received a book and a device. As mentioned above for one book the subjects were asked to use a smart phone application, and for the other book they were asked to use ReachMedia. The subjects performed the tasks, and the time to completion was recorded pre book. The subjects were asked to declare completion when they felt they had a good enough understanding of the book style and content.

After the subjects interacted with the two books they answered a questionnaire in which we asked them to estimate the duration of each interaction, and to answer a few explicit questions about the usability and cognitive load of the interface. The questions were of the form "How obtrusive was the gesture interface in respect to the usual way you flip through and hold a book?" and subjects responded on a 10-point BIDR-style scale that ranged from (1) "not at all" to (10) "Very much". (See Appendix C – ReachMedia Usability Evaluation).

8.5 Result

All the subjects managed to complete the task and listen to the reviews using ReachMedia.

The TLX cognitive load scale results show that the subjects rated ReachMedia as much more mentally demanding than the phone application with the average scores of 3.1 for the phone versus 7.2 for ReachMedia (ANOVA for the condition effect gave $F(2,11) = 4.35$, $p = 0.001$)., which confirms our assumptions about the relative high cognitive load of ReachMedia. The results of the RSD were not statistically significant toward any of the conditions however the averages of the RSD were similar with 1.18 for the phone and 1.21 for ReachMedia, which is an indication that overall in terms of usability and flow the devices performed in a relatively similar way.

We think that this supports out hypothesis that ReachMedia, compensated with its user interaction design for the high cognitive load.

This is also backed by the fact that ReachMedia was much more inviting for the subjects to use in parallel while flipping through the book. Out of the 11 subjects only two flipped through the book while using the phone compared to 8 using ReachMedia. We asked one of the subjects at the end of the task why he did not flip through the book while holding the phone and he remarked, "it just didn't seem possible to flip while holding the phone".

Subject accessed about the same amount of reviews with both devices, however they spent longer time interacting with ReachMedia with an average of 160 seconds compare to 94 seconds with the phone (ANOVA for the condition effect gave $F(2,11) = 4.41$, $p = 0.059$). We assume that this is mainly due to the high cognitive load and the fact that most of the subjects were not familiar with the device and found the orientation and navigation in the menu challenging. Subjects made 2-3 mistakes in average, which led them to visit the same node twice or to enter a review twice. Some occurrences of miss-classifications also contributed in some cases to a disorientation that resulted in some time loss.

Subjects were ask to rate how much they liked each one of the devices, the results show a slight preference to ReachMedia with 6.6 versus 6 for the phone. However this preference was not statistically significant (ANOVA for the condition effect gave $p = 6.5$.

8.6 Informal Observations

The most apparent phenomena we witnessed in the experiment was the subject's difficulty to remember their gesturing pattern as executed while training the system. 6 out of the 11 subjects changed their gesturing style as performed in the training session when actually interacting with the system. When we noticed that during the introduction stage of the experiment the experimenter guided the subjects to try and replicate their gesturing style as used in the training session. We think that the fact that the gestures are relatively small and subtle makes them harder to be repeated and remembered in a precise way. This fact suggests that the gesture recognition system needs not only to learn the user's gestures, but also to allow the user a dynamic and responsive feedback that will help her remember her gestures the next time she uses the system. This might be achieved using feedback that will give the user some sense as to how her gestures differ from the model. Also it suggests that the question of user-dependent versus user-independent model should take into consideration the user's ability to remember their gestures.

An interesting behavior that was observed with a few of the subjects while practicing with the system, was that when the system failed to recognize a gesture, the subjects would sometimes attempt to repeat the gesture with more energy and intent, however, due to the gesture recognition algorithm design, which emphasized noise filtering and a low false positives rate, all those high energy gestures where classified as noise. As mentioned before this loop of misclassification and the subsequent increase in the infliction energy was observed with 4 out of the 11 subjects. In all these cases the examiner intervened and explained to the subject that the gesture should resemble as much as possible the gesture as performed while training the system, which was usually more subtle. The intervention helped all the users to complete the task successfully. This could suggest that the system might need to adapt dynamically, and use features that are scale independent.

Additionally 2 of the 8 subjects that flipped through the book while interacting with ReachMedia received false alarms from the gesture recognition system resulting from the book flipping movements they made.

As to the RFID touch interface, the subject's awareness to the position of the tag on the manuscript encouraged all of them to purposefully touch the tag with the wristband in something that resembles a barcode "scan" motion and none of the subjects really tried to use the device as an implicit interface.

Although the audio interface was not evaluated specifically in this experiment it is important to note that users had a hard time listening to the text to speech audio and many of them complained about it.

8.7 Conclusions

The data and observations resulting from the user study support our hypothesis that the ReachMedia interface enables and supports interaction with just-in-time information about everyday objects.

A few issues that were raised in this user-study are of great significance mainly concerning the gesture recognition system. The system's attempt to perform continuous gesture recognition results in a very narrow band of error for the user, and a highly sensitive classifier. Additionally, the fact that the gestures are very subtle makes it hard for the user to replicate the exact gestures, which were used for training. As a result we get the phenomena described above were the users try using more energy then needed.

We therefore conclude that it might be beneficial to add to ReachMedia a mode in which the user can explicitly declare the beginning and end of gestures, maybe by using a ring like the one used by Marti [Marti 2005].

We also assume that such a configuration will help solving the false positives (false alarms) that were triggered by fine activities such as flipping through a book.

9 Gesture Recognition

There are two approaches to the dynamic gesture recognition problem [Mäntyjärvi 04]: Discrete Gesture Recognition (DSR) and Continuous Gesture Recognition (CGR). DSR uses an explicit command from the user to indicate the start and stop of a gesture via a non-gestural modality, e.g., with a button. With CGR, on the other hand, the recognition is carried out online from a continuous flow of gestural input data.

Naturally, implementing a DSR system requires the user to hold a secondary device, which conflicts with our hands-free design choice. The system therefore was designed as a CGR, which apart of the challenge of correctly classifying gestures, posed two major challenges: first, detecting an activity in the signal flow and second, correctly identifying noise or "unknown" gestures. The latter is especially problematic in a system such as ours because the "unknown class" includes "everything" while at the same time the system requires a low false-positive rate.

9.1 Acceleration Signal Pre-processing

The first step in a CGR system is the segmentation of the acceleration signal into "windows" which will then be fed into the classification algorithm.

Two approached were considered while designing the system, the first approach is the fixed length window approach, which means that the system segment the signal into fixed length windows, usually with a 50% overlap between windows. The second approach is to preprocess the data in order to detect the actual period of activity in the signal and then send only those parts to the classifier. Because our system targets mobile devices and efficiency is a necessity, we chose the latter approach.

We detect activity by using a simple axis-by-axis variance window with preset thresholds using the methodology introduced by Benbasat [2], whose work also focused on mobility. He has shown that the per-axis variance window method is both very effective and is easily optimized for real-time classification. Benbasat's work also indicated optimal start and stop thresholds values as a function of the window size. We have chosen to use a 16-sample window with a start threshold of 100 and a stop threshold of 50. When the variance of one or more axis passes the "start" variance threshold, we begin recording the gesture. We stop recording once the magnitudes of all of the axes are below the "stop" variance threshold. Naturally, this method required the user to have a short pause before and after the gesture. At the current sampling rate, this pause is <100ms per gesture, and generally imperceptible in normal usage.

9.2 Hidden Markov Models

Much of the research involving gesture recognition systems [Westeyn 03, Mäntyjärvi 04], uses Hidden Markov Models (HMMs) for classification due to the signal's temporal character. HMMs are statistical models in which the system is assumed to be a "Markov Process" with visible observation and hidden states. Each state has few probabilities associated with it: a prior probability, transition probabilities to move from it to another state, and the probabilities to produce an observation. HMMs can be either discrete or continuous, in discrete HMMs the observations are a finite set. However in continuous HMMs the observations are real numbers, therefore the probabilities of seeing an observation in a given state is represented by a PDF (Probability Distribution Function).

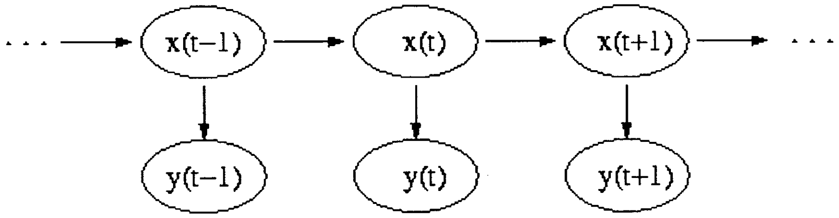


Figure 13: A graphical model presentation of HMM

One of the advantages of HMM is its capability to handle variable lengths of input. Most of the accelerometers based gesture recognition systems known to us take advantage of this capability and use the raw signal vector as an input to the model, or a scalar resulting from vector quantization (over the axis vector) [Mäntyjärvi 04]. However, the problem we wish to solve is subtly different from this general case. First, the gestures we are proposing are very slight and short, and second because we are using CGR, we need to filter out noise with a very low false-positive rate - an issue which most of the literature on the subject does not address.

9.2.1 Preliminary Tests

We performed a preliminary evaluation of the performance of this raw data HMM approach. Using data collected from one user with 197 gestures, with approximately 60 from each type and 96 samples of noise data.

We chose to use a simple left-to-right HMM type of models for the system and tested various states number of states modes in order to find the optimal the number of hidden states. We differentiated between the model of the noise class and the models of the gestures, assuming that the complexity of the noise class is higher than that of the gestures. We tested 40 combinations of models (a 5 by 8 matrix), with the number of states per gesture model ranging from 1 to 5 and the number of states for the noise class ranging from 1 to 8. Each model was tested using 10 fold cross validation.

We found that 3 states for each gesture and 5 states for the noise class gave the best results, with 93% accuracy on the data with 20% of the samples used for testing.

Although these preliminary results used only one subject, we found them sufficient evidence to indicate that the Naïve Bayes was a comparable algorithm and was a better choice for the system given its efficiency, especially due to our assumption of a relative low variance in gestures across users. Also, this assumption was proven to be correct in the user-independent versus user-dependent tests, which we discuss in a following paragraph.

9.3 Proposed Approach - Vector Based Classification

In the next paragraphs we will introduce a "vector based" technique i.e. a technique that classify gestures based on a fixed length vector of features. The motivation for using "vector based" methods arises from the fact that we designed the system with a particular set of gestures in mind and HMM's capability of learning any given model was not required. We therefore wanted to check whether a simpler classifier that used a customized feature vector was capable of competing with the HMM. One of the main disadvantages we find in the HMM approach described above compared with the "vector based" approach, is that the HMM approach incorporates the knowledge about the problem through the structure of the model, however there is no formalized way for choosing a model and the choice of the model dependent many times on trial and error and experience. The advantage of our "vector based" approach is that it allows the embedding of our understanding of the problem through the choice of features, which is usually a more formalized and rational process.

We will first describe the Digital Signal Processing (DSP) algorithms used to extract the features from the raw data and then we will describe the "vector based" classification methods that were considered and tested for this thesis.

9.4 Features Vector

Given a gesture with n samples, where each sample is a vector of length 3, and each element in it represents an axis, we extract the following 25 features:

Length (1) – the number of raw signal vector samples included in the gesture, n in this case.

Power (2-4) – the energy in each axis is calculated according to Parseval's theorem as follows:

$$P(\overline{a}) = \frac{1}{n}\sum_{i=0}^{n-1} a_i^2$$

Cross Correlation (5-7) – The pairwise similarity of the signal on two different axes, as measures using the correlation coefficients which are defined as:

$$\rho(A_i, A_j) = \frac{\operatorname{cov}(A_i, A_j)}{\sigma_{A_i}\sigma_{A_j}}$$

and calculated for the a given gesture as:

$$r(A,B) = \frac{n\sum a_i b_i - \sum a_i \sum b_i}{\sqrt{n\sum a_i^2 - (\sum a_i)^2}\sqrt{n\sum b_i^2 - (\sum b_i)^2}}$$

We calculate r for all the pairwise combinations of axes, in our case: xy, xz, yz.

Turning points (8-25) – The rest of the features are a result of a signal-processing algorithm designed to find the 3 most significant turning points in the signal. The choice of 3 points arises from the observation that our set of gestures has three main velocity change points: start, turn back and stop.

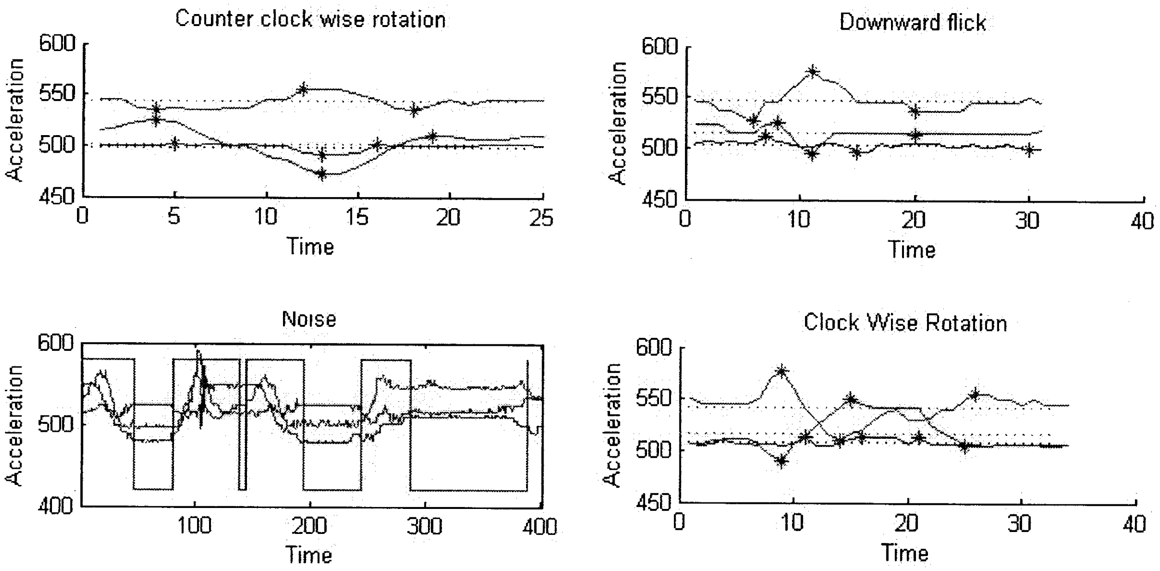


Figure 14 Acceleration signals samples of the 'next' gesture with the turning points marked with '*'

Finding the significant turning points requires us to have a reference by which to measure the magnitude of the signal relative to its starting point i.e. a zero point. The algorithm utilizes some special properties of the signal to do so. Because the gestures start and end at approximately the same point in space and with zero velocity, the static acceleration due to gravitational effects on each axis at the beginning and end of the gesture should be at least approximately equal.

Furthermore, the mean acceleration throughout the gesture should also approximately be equal to the static acceleration value. Therefore, we can use the mean as a reference point for calculating a "relative magnitude" of a sample, which is the difference between the sample value and the gesture mean. The magnitude of the sample is then the absolute value of the relative magnitude. We then choose the three maximal peaks from both sides of the mean (see Figure 14 Acceleration signals samples of the 'next' gesture with the turning points marked with '*)

9.5 Naïve Bayes

The Naive Bayes classifier is a simple probabilistic classifier that can be derived from the Bayes theorem. This model assumes that each feature directly influences the probability of each class and that the features are all conditionally independent. Although this assumption is over-simplified (or naïve), this classifier show many times better results than more complex classifiers and recently there were found some theoretical explanations of its seemingly unreasonable efficiency.

From a graphical model point of view it can be seen as a very simple Bayes Net with one node per class and one node per feature, where the only edges in the graph are edges connecting each feature node to all the classes nodes (see Figure 15: Naïve Bayes graphical model representation).

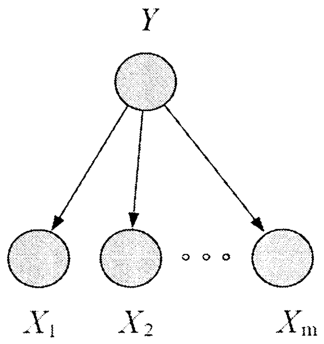


Figure 15: Naïve Bayes graphical model representation

9.5.1 User Study

In order to evaluate the accuracy of the naïve Bayes online gesture recognition system, we trained and tested the system with 10 subjects: 8 males and 2 females, all colleagues. Each subject trained the system for 2 minutes using 60 gestures (approximately 20 samples of each gesture, selected at random by the system). The subjects then tested the resulting model with another set of 30 gestures. The training and testing were done by an automatic system that randomly prompted the users to perform a gesture every 2 seconds.

The results were very encouraging with 4 users having 100% accuracy; and an average accuracy of 94.8% with a variance of 30%.

Since the results above are achieved by using user-dependent models we wished to evaluate what the accuracy cost would be when using a user-independent model. From a user experience point of view, the system is relying on a personal device. Hence, a first time, short training session that takes less than 2 minutes, does not seem unreasonable. Nonetheless it is important to get a measurement of how much more accurate a user dependent model is in order to assess its necessity. In order to get this measurement we used a 5 fold cross validation analysis using all our data. For each fold we trained a model using eight subjects and tested it on the remaining two. The resulting average accuracy was 91.2% with a variance of 10.1%.

	Next	**Previous**	**Select**
Next	0.97	0.01	0.02
Previous	0.04	0.96	0
Select	0.02	0.02	0.96

Figure 16: Between-gesture confusion matrix for 779 gestures samples

We consider these results to be high enough to justify further evaluation of the usability of the user-independent gesture recognition system

9.6 Support Vector Machine

Because we used different types of input for the HMM and the Naïve-Bayes classifier, we wanted to compare the results of the Naïve-Bayes classifier to results from a baseline "vector based" classifier. We decided to use the popular Support Vector Machine (SVM), which is an optimized linear classifier that separates the data into two classes with a maximal margin. SVM can also use the "kernel trick" to transformed the feature space into a non-linear space, so that the classifier is still linear in the resulting non-linear feature space however it is non-linear in the original feature space. SVM is a very powerful classifier, which requires a lot of resources for training but is relatively lightweight in real time. Once the training is done the SVM is using only a fraction of the data - called "support vectors" - which make it an efficient algorithm at run time.

9.6.1 Kernel and Kernel's Parameter Selection

Tests were done with few kernels and cross-validation was used for Kernel's parameter selection.

The first Kernel that was tested was the Gaussian (RBF) kernel. In this kernel, there are two parameters. First, there is the usual SVM error penalty parameter C coming from the SVM optimization problem:

$$\min C\sum_{i=1}^{l}\xi_i + \frac{1}{2}\|w\|^2$$

The second parameter is coming from the Gaussian Radial Basis Function:

$$\text{for } r = \|x\| \;\; K(r) = e^{r^2/2\partial^2}$$

LIBSVM's grid script was used to check the parameters in the following ranges:

$$3 > \log_2(\partial) > -3, 3 < \mathrm{C} < 3$$

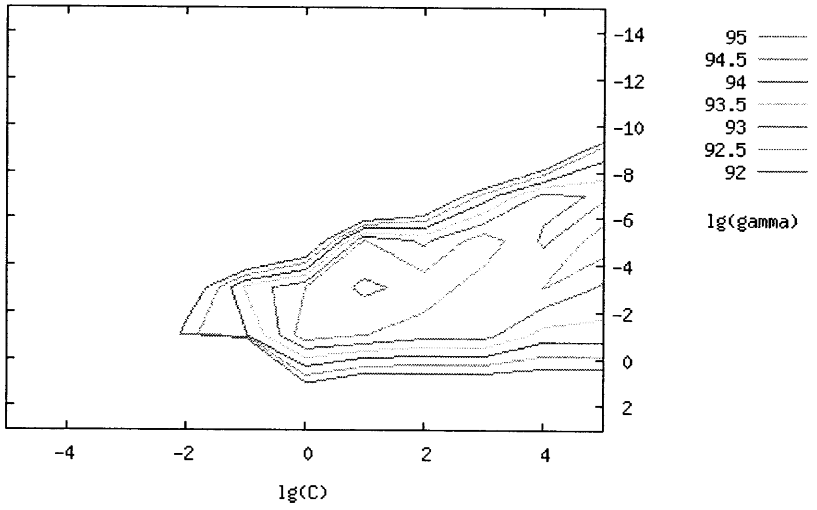


Figure 17: RBF kernel parameter selection (Gamma and C) using LIBSVM cross validation

The script constructed a 10*10 grid of all possible combination within the above range. For each combination, a model was constructed and tested. The results are plotted in Figure 17. As can be seen 3>C>5 and 0.01>sigma>0.001 are giving the best result at a rate of about 95%.

Linear SVM was also tested and compared to the RBF results. We have found that the results of the Linear SVM where equivalent to the RBF's accuracy rate using the same C value.

9.7 Gesture Training Test

Another important parameter in the evaluation of the system is the effect of the number of training examples on the classification rate. To ensure a good user experience the system must be trained using the minimal number of samples. Ideally we want our system to be trained with not more than a couple of dozens examples per gesture, in an overall process that will take no longer than one or two minutes.

In order to understand better the effect of the number of samples on the classifier accuracy we created eight subsets of the training data ranging from 12 – 90 samples per gestures. We trained and tested the resulting models using the remaining 20% of the data. As can be seen in Figure 18: SVM model accuracy as a function of the number of training samples, we start getting good results with around 13-20 samples per gestures. And the Naïve Bayes performed better then the SVM regardless of the number training samples.

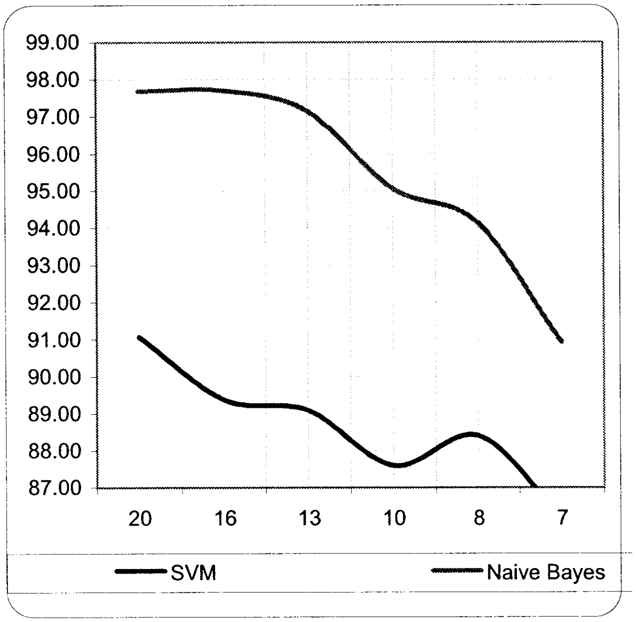


Figure 18: SVM model accuracy as a function of the number of training samples

10 Discussion and Future work

In this thesis, we presented a holistic approach to the problem of on the move interaction with augmented objects, integrating solutions from user input, to processing of input, to user output. On the move interactions present a completely new class of interaction issues, and thus require a comprehensive approach. Our work thus far has been focused on the technical implementation and development of the hardware, gesture recognition software, RFID antenna and context manager. We now have a working system with reasonable RFID range and very reliable gesture recognition.

We have evaluated the gesture recognition algorithms and showed that they are able to classify with a high enough recognition rates for the needs of the system.

We also conducted user studies that have shown that the system was usable and was able to deliver information to the user while keeping the user in the flow of their activity.

The user's studies also raised few issues concerning the gesture recognition interface, in particular regarding the design choices of the gesture recognition system. We believe that further investigation of directions such as dynamic adjustment of the system and improvements of the feedback to the user could be an interesting path for future research.

Additionally, improvements to the RFID reader's range and further evaluation of the touch based interaction are still to be thoroughly investigated.

11 References

G. Abowd, C. Atkeson, J. Hong, S. Long, R. Kooper, M. Pinkerton, “Cyberguide: A mobile context-aware tour guide”. ACM Wireless Networks, 3:421--433, 1997.

A.Y. Benbasat and J.A. Paradiso “Inertial Measurement Framework.” International Gesture Workshop, GW 2001, London, UK

T. Baudel and M. Beaudouin-Lafon. “Charade: remote control of objects using free-hand gestures”. Communication of the ACM, 36(7): 28–35, July 1993.

B. B. Bederson, Interfaces for staying in the flow. Ubiquity 5, 7 (2004).

S.A Brewster, J. Lumsden, M. Bell, M. Hall and S. Tasker “Multimodal 'Eyes-Free' Interaction Techniques for Wearable Devices”. CHI 2003.

G. Chen and D. Kotz. “Context aggregation and dissemination in ubiquitous computing systems”. Workshop on Mobile Computing Systems and Applications. IEEE Computer Society Press, June 2002.

M. Csikszentmihalyi (1991). Flow: The Psychology of Optimal Experience. HarperCollins.

M. Czerwinski, E. Horvitz and E. Cutrell (2001). Subjective Duration Assessment: An Implicit Probe for Software Usability. (Human-Computer Interaction (IHM-HCI 2001)) pp. 167-170.

N. Cohen, A. Purakayastha, J. Turek,, L. Wong, and D. Yeh “Challenges in flexible aggregation of pervasive data”. IBM Research Division, Thomas J. Watson Research Center. Technical Report RC21942 2001.

E. Costanza, S. A.Inverso, R. Allen. “Toward Subtle Intimate Interfaces for Mobile Devices Using an EMG Controller”. CHI 2005, April 2005.

C. Florkemeier, M. Lampe “Issues with RFID usage in ubiquitous computing applications”. Pervasive 2004.

K. Hinckley, J. Pierce, E. Horvitz, and M. Sinclair. “Foreground and Background Interaction with Sensor-Enhanced Mobile Devices”. ACM Transactions on Computer-Human Interaction (TOCHI) 2005, 12(1), 31-52.

R. Hull, P. Neaves, and J. Bedford-Roberts “Towards Situated Computing” Proc. ISWC 1997

P. Lukowicz, J.A. Ward, H. Junker, M. Stäger, G. Tröster, A. Atrash, T. Starner: “Recognizing Workshop Activity Using Body Worn Microphones and Accelerometers”. Pervasive 2004.

J. Mäntyjärvi , J. Kela, P. Korpipää, S. Kallio, “Enabling Fast and Effortless Customisation in Accelerometer Based Gesture Interaction " Mobile and Ubiquitous Multimedia, Maryland, USA, Oct. 27, 2004.

E. Munguia Tapia, N. Marmasse, S. S. Intille, and K. Larson, "MITes: Wireless portable sensors for studying behavior," Ubicomp 2004.

M. Philipose, K. Fishkin, D. Fox, H. Kautz, D.J. Patterson, M. Perkowitz. “Guide: Towards Understanding Daily Life via Auto-Identification and Statistical Analysis”. UbiHealth 2003.

A. Pirhonen, S. Brewster, and C. Holguin. “Gestural and audio metaphors as a means of control for mobile devices”. CHI, 2002. Minneapolis, Minnesota, USA

J. Rekimoto “GestureWrist and GesturePad: Unobtrusive wearable interaction devices”, Proc. ISWC 2001, 21—27

Sadi, S. Maes, P. "xLink: Context Management Solution for Commodity Ubiquitous Computing Environments." In UBICOMP Workshop ubiPCMM: Personalized Context Modeling and Management for Ubicomp Applications. Tokyo, Japan, 2005.

N. Sawhney and C. Schmandt, "Nomadic Radio: Speech & Audio Interaction for Contextual Messaging in Nomadic Environments," ACM Transactions on CHI, vol. 7, pp. 353-383, 2000

T. Starner, S. Mann, B. Rhodes, J. Levine, J. Healey, D. Kirsch, R. Picard and A. Pentland, "Augmented Reality Through Wearable Computing", Presence 386-398 Winter 1997

D. Wan "Magic Medicine Cabinet: A Situated Portal for Consumer Healthcare" in Proceedings of First International Symposium on Handheld and Ubiquitous Computing (HUC '99), September 1999.

R. Want, K.P. Fishkin, A. Gujar, and B. L. Harrison “Bridging Physical and Virtual Worlds with Electronic Tags”. In Proc. of CHI 1999

R. Wasinger, A. Krüger. “Integrating Intra and Extra Gestures into a Mobile and Multimodal Shopping Assistant”. Pervasive, 2005.

T. Westeyn, H. Brashear, A. Atrash, and T. Starner. "Georgia tech gesture toolkit: supporting experiments in gesture recognition." Conference on Multimodal interfaces, 2003.

12 Appendix A – data flow in gesture recognition system

This is a UML sequence diagram showing the data flow between the gesture recognition software components while handling the incoming signal data, from the feeder all the way up to the classifier. The time flow in UML sequence diagrams is from top to bottom. Please note that the diagram was divided into two in order to fit it into the page the origal diagram is a horizontal concatenation of the two diagrams.

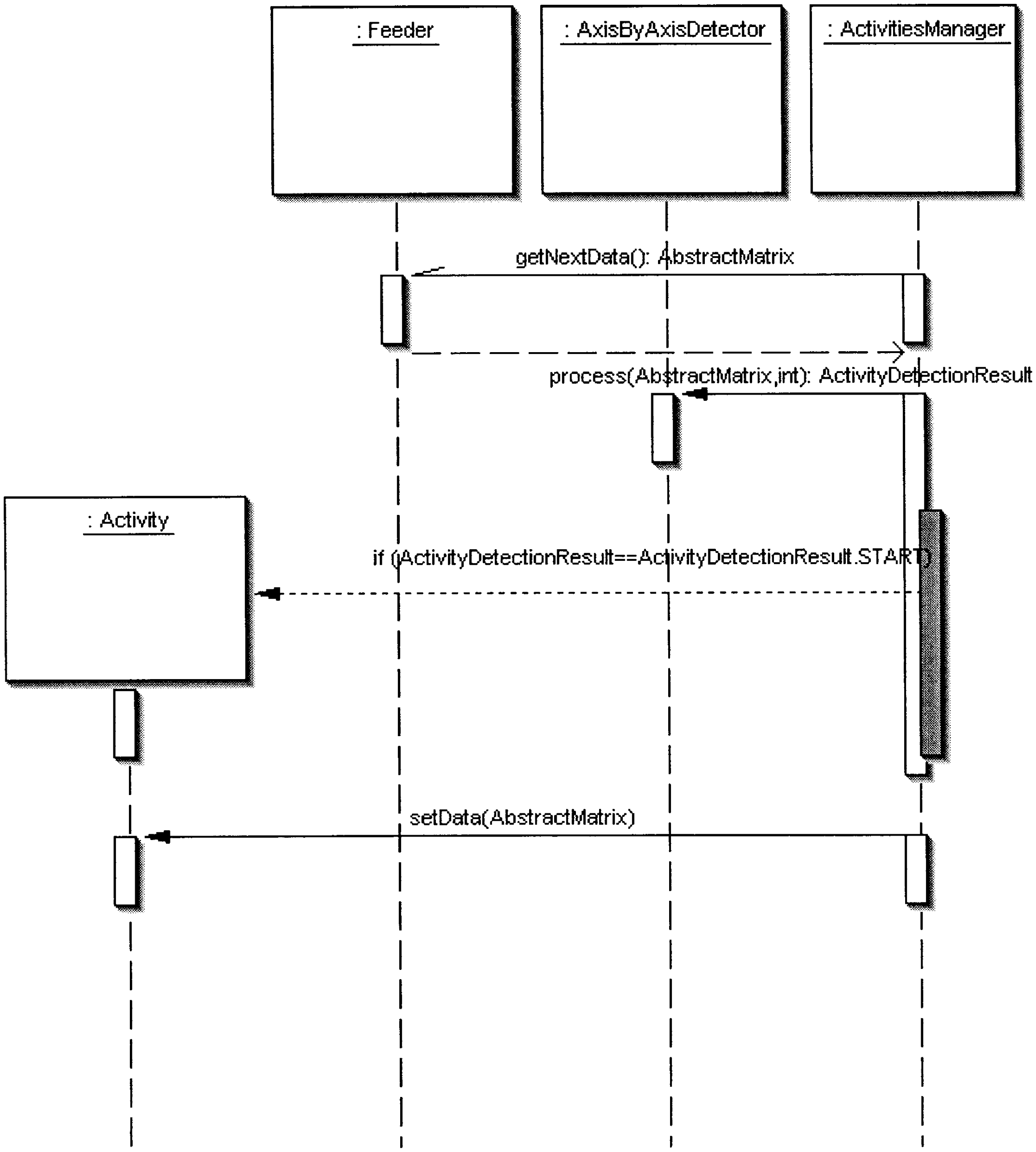

: Feeder
: AxisByAxisDetector
: ActivitiesManager
getNextData(): AbstractMatrix
process(AbstractMatrix,int): ActivityDetectionResult
: Activity
if (ActivityDetectionResult==ActivityDetectionResult.START)
setData(AbstractMatrix)

Figure 19: First sequence

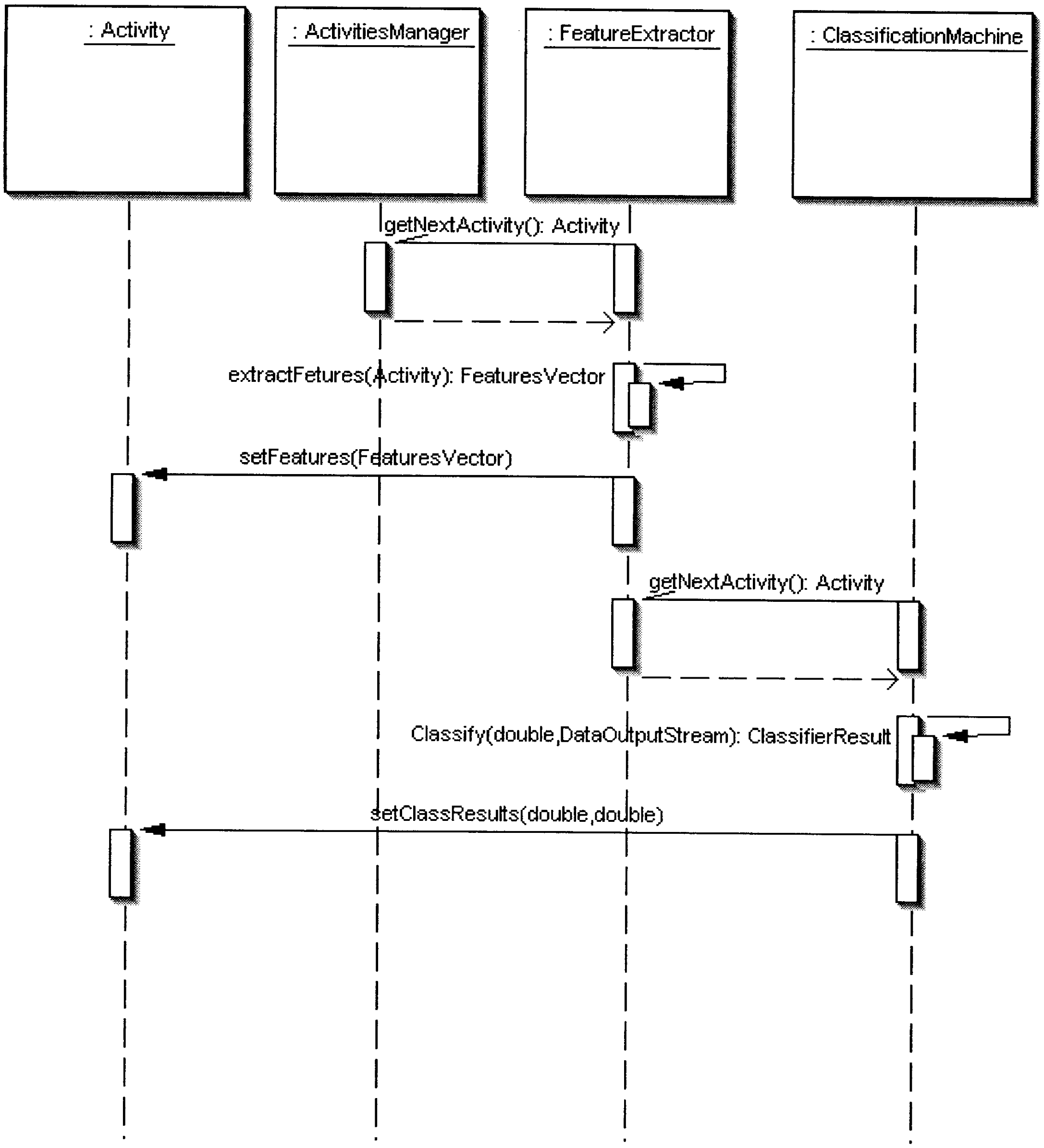

: Activity
: ActivitiesManager
: FeatureExtractor
: ClassificationMachine
getNextActivity(): Activity
extractFetures(Activity): FeaturesVector
setFeatures(FeaturesVector)
getNextActivity(): Activity
Classify(double,DataOutputStream): ClassifierResult
setClassResults(double,double)

Figure 20: Sequence continued…

13 Appendix B – NASA TLX

Please place a mark on each scale that represents the magnitude of each factor in the task you just performed

Mental Demand: How much mental and perceptual activity was required (e. g., thinking, deciding, calculating, remembering, looking, searching, etc.)? Was the task easy or demanding, simple or complex, exacting or forgiving?

Very Low Very High

Physical Demand: How much physical activity was required (e. g., pushing, pulling, turning, controlling, activating, etc.)? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?

Very Low Very High

Temporal Demand: How much time pressure did you feel due to the rate or pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?

Very Low Very High

Own Performance: How successful do you think you were in accomplishing the goals of the task set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing the goals of the task set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals?

Very Low Very High

Frustration: How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

Very Low Very High

Effort: How hard did you have to work (mentally and physically) to accomplish your level of performance?

Very Low Very High

Annoyance ???

Very Low Very High

14 Appendix C – ReachMedia Usability Evaluation

Please estimate how long it took you to make up your mind about the book

____________________.

Please place a mark on each scale that represents the magnitude of each factor in the task you just performed

1. Did you like the system?

   Not at all Very much

2. How helpful was the system in the decision making process?

   Not at all Very much

3. How obtrusive was the interface in respect to the usual way you flip through and hold a book.

   Not at all Very much