Abstract

Wireless communications are important in biomedical applications to measure patient vital signs and analyze patient behaviors and conditions. Using small sensors that are worn on a patient’s body, it is possible to obtain various data values, such as kinematic and vital sign data, to monitor patient activity and conditions. As the sensors are small and constantly send wireless data, it becomes possible to monitor patients unobtrusively and in real-time. The kinematics data from the sensors is used to investigate behavior of smokers and their frequency of smoking, as well as the potential implementations of other sensors that monitor vital data, such as heart rate and skin resistivity.

1. Introduction

There has been a rapid increase in the use of wireless sensors in the medical field in order to monitor patients in real time. These sensors are deployed on a patient to form a cluster called a Wireless Body Area Network (WBAN), which monitors physical conditions and cooperatively passes its data through the network to a central location [1].

Outside of specific body area sensor systems, data collection is very vital for medical operations, as objective numeric data needs to be collected on patients to monitor their vital signs and diagnose diseases. Such information is particularly useful in emergency scenarios when efficient patient assessment is crucial to providing proper care and maximizing appropriate resources [2]. To monitor and obtain patient data, hospitals and doctors all over the world use a variety of software and hardware systems. The information is then analyzed to obtain crucial information about patient condition.

The analysis process would be expedited by the integration of accelerometer measurements and vital sign recordings, as the former is instrumental to the understanding of the basis of vital signs themselves. The kinematic information obtained from the accelerometer serves to establish the framework and context for observing patient conditions, such as the type of
activity that is being performed and how these movements affect vital signs.

Despite some benefits, current methods of collecting data for medical purposes are problematic for collecting ongoing vital readings. The most prolific data collection models used in hospitals today consist of static, one-time tests that may not accurately portray patient conditions. In addition to the high cost, the main problem with current methods used for collecting data outside of the hospital, such as the use of high-level cameras to monitor movement, is that they don’t allow freedom for the patient. The patient is required to be in a fixed position for the cameras to work effectively. The use of WBANs to collect data can provide useful information about patient conditions and portray a more accurate depiction of vital signs while allowing the patient to live their normal lives. Because the sensors are wireless, the person is free to move while the constant monitoring of the vital signs is taking place.

Since WBANs operate in real-time, vital signs are collected over a period of time, which allows for the tracking of changes that static tests would otherwise overlook. For example, blood pressure measurements are typically assessed through pressure indicators along the upper arm. This method, however, does not sufficiently take random outliers into account, particularly in other parts of the body, and as a result, the acquired data may be skewed. Using wireless sensors that continuously and extensively evaluate vital signs will address such inadequacies in reviewing patients’ conditions [3].

In addition, WBANs have the other advantage of seamlessly providing objective data to generally subjective subjects, such as sleep quality. Sleep is a common method for monitoring the effectiveness of medical treatment.
Current methods of analyzing sleep patterns are almost all subjective (questionnaires, etc.), which produces a bias. A previous study entitled “Night Owl: A Quantitative Sleep Journal,” measured heart rate and skin conductance data to quantitatively distinguish the level and depth of sleep using a Shimmer mote, which is a small sensor that is portable, wearable, and can transmit data wirelessly. Shimmer is the main component required to set up a wireless sensing application, and it provides a range of functions including basic motion sensing, data capture, processing, and the wireless transmission of sensed data [4]. It then uses Bluetooth to transfer the data from the motes to a computer where software is used to further analyze the data [5]. By monitoring various vital signs using the sensor, it is possible to track patterns and record changes in vital signs over time to find relationships between vitals and biological phenomena objectively.

Another previous project used WBANs to prioritize the transmission of patient vital signs using wireless body area networks as a solution for in-hospital triage (the process of prioritizing patients based on the severity of their conditions) [2]. Thus, WBANs are ideal for both in-hospital and out-of-hospital situations.

2. Background

The goal of our project was to map the vital signs of a smoker by utilizing the functions of wireless sensors and the complementary software programs in order to compare and analyze kinematic data to monitor and predict movements of a patient. By using acceleration data from wireless sensors, we predicted the movements of a patient’s arm to detect smoking behavior. After collecting data, we used data visualization, a method of modeling raw data using graphical representations, to provide a further understanding of the patient’s condition.

We used several tools and methods to collect and visualize the data. The physical sensors used to collect data are Shimmer motes, which run small programs through its native operating system TinyOS. The programming language NesC was used to write the programs for the Shimmer mote because NesC had packages to read data from the sensors and compiled natively on TinyOS. TeraTerm, a client program that reads data through virtual serial ports, was set up to receive the data on the recipient computer. A program called Com2Com created the virtual serial ports, which are what the Shimmer mote connected to via Bluetooth. Then, we used Shimmer Connect, a program on the computer that is linked directly with the wireless sensor, which graphs the resultant data in real-time. Finally, through the use of Unity 3D, a game development kit, we were able to create a visual representation of the data we received from the Shimmer motes.

2.1 Shimmer Mote

The Shimmer motes are small, programmable sensors that measure 53mm x 32mm x 15mm. Each mote contains its own Central Processing Unit (CPU), which runs the TinyOS operating system, three programmable LEDs, a MicroSD card slot, and a Bluetooth communications module. In addition to those core components, the sensors used also contain accelerometer and gyroscope sensors that output data to the internal memory. Both the accelerometer and the gyroscope output data in three
axes, with each axis corresponding to a rectangular face of the Shimmer mote.

The code for the Shimmer mote was written in NesC, a C-based programming language, which was compiled and run directly on the TinyOS operating system. We used two programs in this project – one, SimpleAccel, took the accelerometer data from the sensor and outputted it over Bluetooth as 16-bit integers, while the other, AccelGyro, exported both accelerometer and gyroscope data in packets that could be read by the receiving computer [4].

2.2. Teraterm and Com2Com

Teraterm is a terminal emulator, which opens the fundamental serial port that connects to the Unity 3D program. We scripted Unity to access the serial port through Teraterm so the compiled data can be effectively relayed between the sensors and the corresponding visualization medium. This feature was precipitated by the incorporation of Com2Com, which sets up the port prior to configuring Teraterm.

Com2Com is a virtual serial port program that allows us to connect various ports to whether or not the sensor is able to connect to the computer. The serial port is an interface that can transmit one bit at a time for serial communication (unlike a parallel port, which can receive and transmit data bits simultaneously) and can be connected to modems, printers, etc. In this project, we used serial ports to transfer the data from Bluetooth to our programs.

Com2Com was primarily used to launch individual serial ports that would ultimately connect to the host computer and exchange sensor reports for further examination. In order to accomplish this, we initially prepared the program to access two ports, which then allowed us to link the opened serial ports to other systems.

2.3. Shimmer Connect

In order to plot the raw data from the Shimmer mote, we used an application called Shimmer Connect, which allows users to display and save data received from Shimmer devices streaming over
Bluetooth. The application is designed for greater usability and functionality [6]. It plots the x, y, and z axes of the data, so it enabled us to visualize how the data was changing when the mote was positioned differently, moved at slower/faster paces, and so on.

2.5. Unity 3D

Unity 3D is a game development kit that contains graphics and physics logic, and is scripted in C#. We used the game engine to visualize the data and create a model of the acceleration values.

3. Process

The course of our project development relied on a combination of components and tasks, ranging from basic port communication to more intricate data analysis. At its core, the procedure applied individual computer programs that carried out specific purposes within the overall structure of the project. The Unity 3D game engine and specific C# scripts were used to process the transmitted information to effectively model the patient’s actions. The activity of a smoker can be subsequently outlined and expressed by the data-visualization software. Furthermore, this information can be used to understand the patient’s situation while vital signs are being collected.

The monitoring of the patient starts with the Shimmer mote where the accelerometer is used to measure the changes in speed over time of the location that the sensor is placed. The raw data from the sensor is then sent via Bluetooth to Teraterm. The data on Teraterm is then passed to the Unity 3D program, which is employed to design a visualization process that models the raw data obtained from the Shimmer mote in real-time.

We designed and programmed an arm using the packages available on the software program to move in response to our data. Using Unity, we also wrote a script to allow the data to be translated from raw accelerometer data into integers that could be applied to the arm visual in the Unity program. We programmed the model of the arm to respond to data and simulate the movement of the smoker. Unity 3D was
then connected to Teraterm through the application of a serial port communication script. The script allows information to be directly transmitted between the programs and arm movements. The Unity program obtained data from the Bluetooth connection via this serial port.

3.1. Application of Software

To organize an efficient operation for carrying out our objectives, we adopted a specific approach to address our various tasks. Our primary targets centered on the development of the sensor program as well as the transfer and analysis of the resulting figures. To accomplish this task, the overall process was split into two parts: the first was programming the sensor to recognize hand gestures and store data based on the motions, and the second was designing an arm visual program to receive, interpret, and display the data extracted from the sensor.

4. Data Collection

Data was collected directly from the Shimmer mote, and recorded via the NesC programs written onto the device. The program worked by establishing a Bluetooth connection between the Shimmer mote and the host computer, and then sending packets containing accelerometer data from the sensor to the computer’s serial port.

The individual data packets sent from the sensor were single-byte integers with each byte representing X, Y, or Z data in that order. To avoid sending too many packets to the host computer at any one time, two arrays were set up as buffers to hold data. At any one point, only data from one of the

```c
async event void DMA0.transferDone(error_t success) {
    atomic DMA0DA += 8;
    if(++dma_blocks == 15)
    {
        dma_blocks = 0;
        if(current_buffer == 0){
            call
            DMA0.repeatTransfer((void*)ADC12MEM0_,(void*)sbuf1,NBR_ADC_CHANS);
            current_buffer = 1;
        }
        else {
            call
            DMA0.repeatTransfer((void*)ADC12MEM0_,(void*)sbuf0,NBR_ADC_CHANS);
            current_buffer = 0;
        }
        post sendSensorData();
    }
}
```

Transferring data to ADC and swapping buffers
buffers was sent to the computer via Bluetooth while data was written from the sensor into the other buffer. At any one point, only data from one of the buffers was sent to the computer via Bluetooth, while data was written from the sensor into the other buffer. This ensured that no data was overwritten before it was sent out, and kept operations smooth without jumps or lags. At certain points in the program, the buffers were swapped to make sure all of the sensor values reached the host computer, and to clear out old data.

The Shimmer mote recorded acceleration values as analog data, but packets needed to send digital data over to the computer. To rectify this issue, we needed to use an ADC (Analog-Digital Converter) that was loaded into the Shimmer mote, to convert the analog values to digital numbers before sending all the data over to memory to be packaged and transmitted over Bluetooth.

After each transfer, the buffers were sent to the ADC (analog-digital converter), and the current buffer was swapped before sending data to Bluetooth via the sendSensorData() task.

The sendSensorData task wrote the entire buffer to Bluetooth, where it was sent wirelessly to the host computer. If the sensor was capable of sending data, the program would check to figure out which buffer was not being written into and send the data from that buffer. After the data was sent out via Bluetooth, the sensor disabled sending out more data until Bluetooth was finished writing and sending the previous set of data.

```c

task void sendSensorData()
{
    atomic if(enable_sending) {
        if(current_buffer == 1)
            call Bluetooth.write((uint8_t *)sbuf0, 120);
        else
            call Bluetooth.write((uint8_t *)sbuf1, 120);
        atomic enable_sending = FALSE;
    }
}
```

The current buffer is sent to Bluetooth if the program allows communications to occur.

The SimpleAccel code works by first establishing a connection to a computer with Bluetooth, and then regularly collecting and sending acceleration data over the Bluetooth connection. When Bluetooth connection is lost, all of the sensors are shut down, and data is no longer collected nor sent.
5. Data Visualization

After the data was sent over to the computer, the data was processed to establish a link between the acceleration and smoking behavior. To begin, a visual analysis of the data was conducted in Shimmer Connect to find characteristic patterns in the acceleration values that corresponded with smoking behavior. To do this, we first assumed that the initial position of the sensor and the arm was in a loose, flat position near the hip, with the Y values pointed down at the ground, as such:

After the Shimmer mote was placed on the arm, we observed visual data from the accelerometer onto the graphs of Shimmer Connect. We found that the y-axis acceleration increased or decreased dramatically whenever the arm was raised or lowered, which in certain situations, can be assumed to be indicative of smoking behavior.

A visual representation to receive, analyze, and translate the data into actions was written in C# to interface with the Unity 3D engine. We designed an arm and wrote a script in Unity 3D to program this visual representation to move after reading the data transferred from the Shimmer mote to Unity. We then set up a program called Teraterm, a terminal emulator that opens a serial port and allows the Shimmer mote to connect to the Unity 3D program. We scripted Unity to access the serial port through Teraterm to transfer the compiled data from the Shimmer mote to the Unity program that hosts the arm visual. This allowed the data to be directly transmitted from the accelerometer to the arm to model exactly what the data represents.
The Unity program on the computer searches for the Shimmer mote when it is first opened, and tries to establish a connection to the sensor through a serial port. The program then reads acceleration data from the Shimmer as integers before calculating arm position from the y-axis acceleration values.
6. Results and Discussion

All in all, we were able to successfully model the movement of the arm of a person using the Shimmer mote and subsequent programs/software. This model could prove beneficial to a smoker in order to gain knowledge of how often they smoke, and to doctors who need to know the context of the situation of the patient.

By using the accelerometer to collect kinematic data of the patient’s movements, the specific conditions of the patient became easier to assess. We discovered that whenever the patient moved the arm attached to the sensor, the graphs of the y-axis and z-axis exhibited drastic acceleration peaks and declines. We were able to utilize the sharp changes in the y-direction of the accelerometer in order to allow the Unity model to register the activity of the patient’s arm.

Not only did the accelerometer data prove to be a central component of our procedure, but data visualization also played a large role in our research. The Unity representation provided a model that successfully replicated the patient’s movements as perceived by the sensors. By integrating Bluetooth connections to link to the computer visualization, we were able to create a virtual arm that would mirror the patient’s actions, such as smoking.

Data collection and visualization using the proposed method will allow for wireless, real-time, and continuous data collection and visualization with the right accelerometer graphs and the vital sign graph to the left. Accelerometer graphs depict significant spikes in the y and z-axes when the arm moves up and down.
collection, as well as context awareness.

6.1. Wireless, Continuous, and Real-Time

The wireless sensor allows for real-time detection of the patient’s activity and vital signs. Therefore, it will avoid continuous trips to the hospital, whether for routine checkups or for common medical examinations. By lessening the frequency of hospital excursions, medical centers will be able to decrease the burden and potential complications arising from overextended responsibilities. At the same time, there will be a greater availability of useful resources within medical units because patients can access their conditions from external environments and spare hospitals unnecessary discontinuities in their treatment processes. In other cases, precise measurements can be recorded by the sensors to deliver valuable and relevant guidance to patients.

6.2. Context Awareness

Ideally, the vital sign sensors and the accelerometer/gyroscope sensors will work hand-in-hand. While vital signs are being collected, doctors will be able to simultaneously know the context of the situation; for example, they will know if the patient is running, sitting, smoking, etc. By knowing the context, we are able to gain a greater knowledge of the patient by combining the vital sign data with the situation that the patient is in at that time. This mutual relationship allows the patient to carry on with their lives while being monitored real-time. At the same time, by understanding viable conditions and correlative circumstances, we can assess the potential medical implications associated with the collected spectrum of data.

7. Conclusion

Wireless sensors have a wide array of potential applications, some of which can be explored in future work. Examples of other applications include product development and wireless sensor peripheries, rehabilitation, sports science and athlete development, teaching and university programs, ambient sensing solutions, remote patient monitoring and assistive technology, environmental sensing solutions, market research and biofeedback, and biomechanics [4]. For example, work is currently being done to make games using the sensor for rehabilitation purposes to motivate patients to perform the repetitive movements assigned to them.

These sensors monitor the patient continuously and in real-time. While there are far more precise and accurate methods of monitoring people and tracking signs, these wireless sensors are...
far more portable and do not require large amounts of set-up. Thus, they are potentially more economic for everyday use.

Even small quantities of data, such as slight variations in the extracted reports, can be advantageous to patients who need to monitor small offsets in their body systems. For instance, specific peculiarities in the nervous system can be measured by advanced sensors and used to distinguish smoking, stress, and many other physical conditions [5]. As advancements are made to improve the accuracy of the process, the sensors can be employed to effectively characterize illnesses and sentient states based on vital sign readings.

In theory, these sensors could be used for a vast variety of purposes, ranging from medical to environmental fields of interest. Shimmer motes, with their small, compact structure and integrated 3-dimensional accelerometer sensors, are best suited for wearable applications, such as health monitoring, and can be used when low cost is a concern [1]. Shimmer enables research and industry to be at the top of sensing technology, and it will continue to be useful in various different fields in the future.

8. Acknowledgements

We would like to acknowledge and extend our gratitude to some individuals and organizations. We would like to thank our mentor Dario Pompili for his valuable guidance and instruction. We would also like to thank our advisors Cagdas Karatas and Parul Pandey for fostering our progress through their continuous assistance. As well as, our RTA Amanda Rumsey for her constant support and encouragement. Furthermore, we would like to acknowledge the following sponsors without whom none of this would be possible: Rutgers University, The State of New Jersey, Morgan Stanley, Lockheed Martin, South Jersey Industries, Inc., and PSE&G. And finally, The NJ Governor’s School of Engineering and Technology, as well as its sponsors, for providing us with this monumental research opportunity.

9. References


