ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 13 (2011) 464–469

Engineering Procedia

5th Asia-Pacific Congress on Sports Technology (APCST)

Real time data streaming from smart phones

David Rowlands[a*], Daniel James[ab]

[a]Centre for Wireless Monitoring and Applications, Griffith University, Nathan Campus, Brisbane, 4111, Australia
[b]Center of Excellence for Applied Sports Science Research, Queensland Academy of Sport, Brisbane,Australia

Received 13 April 2011; revised 14 May 2011; accepted 16 May 2011

Abstract

There are many aspects of the athlete's performance that can and need to be measured to improve performance or to fine-tune skills. This can be done visually by a coach or by using sensors attached to the athlete. This paper discusses the common sensor data capture methods and the main requirements for a data capturing system with special attention paid to the requirements for a real time data capturing system. An implementation of a real time data capture system consisting of an iPhone and a laptop is presented. The system was tested on a runner performing a slow jog. The acceleration signatures for the running was streamed from the iPhone to the laptop and displayed in real time thus validating the system.

Selection and peer-review under responsibility of RMIT University

Keywords: Sensor; data capture; real time; iPhone; accelerometer

1. Introduction

There are many aspects of the athlete's performance that can and need to be measured to improve performance or to fine-tune skills. This can be done visually by a coach or by using instrumentation attached to the athlete. In the case of instrumentation, the type of sensors used, and the type of analysis will be different for different sports and athletes. [1-8]. There are many types of sensors available to the sports scientist and coach to use such as GPS for location and speed, inertial sensors for activity monitoring, and other physiological sensors. The advantage of using sensors is that it takes the monitoring out of the laboratory situation and into the field. This makes for a more natural environment for the athlete.

* Corresponding author. Tel.: +61-7-37355383; fax: +61-7-37355384.
E-mail address: d.rowlands@griffith.edu.au.

doi:10.1016/j.proeng.2011.05.115

What is required is a method that will allow the capture, integration, storage and analysis of the sensor based information irrespective of the different sensor units connected to the athlete. This paper gives the requirements for a sensor data capturing system. The common sensor data capture methods are discussed along with a discussion of the main components for a data capturing system. Special attention is paid to the requirements for a real time data capturing system. Finally an implementation of a real time data capture system consisting of an iPhone (inertial sensor monitoring) and a laptop is given.

2. Sensor Data Capture

In a typical athlete monitoring scenario, the athlete has sensors strategically placed on the body to capture the signals of interest. There may be one or more sensors of varying types used concurrently. The athlete would then undergo the action of interest whilst the data is captured. There are two ways that the data can be extracted from the sensor unit to the computer for analysis purposes;

- Wired. Using a static cable downloading the data after the capture.
- Wirelessly. Downloading the data during the capture.

The communications with the sensor depends heavily in upon the type of connection used. This section will examine the communications for both the wired and wireless connections.

2.1. Wired Connection

The wired connection involves the use of a cable to connect between the sensor units and the computer. Typically this is done with the data being downloaded after the capture session has been completed. The most commonly used wired connections are:

- Serial (RS232).
- USB.
- Ethernet.

The Serial connection is the most common connection type especially among the older style sensors. The data is serialized before being sent to the computer [9]. The transmission rates are also not particularly high in the wired version and can subjectively take a long time to transfer large amounts of data. The serial connection is just the physical method of transfer. A protocol needs to be imposed between the computer and the sensor unit so that the data transmission can be initiated, captured and terminated. This protocol is usually specific to the manufacturer of the sensor unit and relies upon the software supplied by the sensor unit manufacturer for data extraction. It should be noted that an adaption of the serial connection has been used in both USB and wireless connections.

USB is a high speed serial connection that is more commonly used to connect between the sensor and the computer. The connection is much more standardized than the typical serial connection especially in terms of the physical connectors. The USB port is hub based which means that a single connection can be expanded to be multiple connections. This hub based structure means that USB can have up to 127 units connected at once each running with transfer speeds of greater than 50 Megabytes per second for USB 2 and above [10-11]. Since there are many different types of USB devices then each device must supply a driver to communicate with computer's Operating System. When the USB device is attached it communicates with the Operating system and tells the Operating System what type of device it is and the Operating Systems chooses the appropriate driver from the list of drivers it has available. The driver is responsible for the protocol of the data transfer between the sensor unit and the computer system. The simplest method for the data transfer is where the USB implements a protocol such that it appears to be a serial device but runs at much faster transfer rates that the straight serial connection. This means that the device appears like it has a serial connection and can be accessed like a serial port which simplifies the programming and access.

Another commonly used method is the Ethernet connection. Like USB this is well defined and the physical connection is also well defined. It operates on a number of layers for communication to occur. In the simplest form it has a hardware connection, with associated protocols superimposed on it (transport layer). The application programs can then send the data in whatever format it wants over this connection. The most commonly used protocols in the transport layer are the TCP/IP and UDP protocols [11]. These protocols are crucial to the way that data is received and impacts upon the reliability of the data. These protocols break the data into packets and communicate by the transmission and reception of these packets. The difference between the protocols is how these packets are handled. In TCP/IP the packets are sent and then an acknowledgment is expected from the receiver. If an acknowledgement is not received then the packet is resent. In UDP, the packet is sent without expecting an acknowledgement which means that if a transmitted packet is lost then it will not get resent leaving a hole in the data received by the receiver. Therefore TCP/IP can be used in transmissions with a more unreliable connection than UDP. In a wired connection the quality of the connection is generally not a problem but becomes more of a problem in wireless transmission as discussed in a later section. It should also be noted that the UDP protocol is a faster transmission protocol than the TCP/IP since it is not constantly processing acknowledgements and that the programming methods of the UDP protocol is simpler than the programming methods of TCP/IP protocol.

2.2. Wireless Connection

The wireless data capture has the advantage over wired connections that no cables are used. This means that the data can be captured as it is collected in real time. This is advantageous since the data can be inspected immediately and any errors with monitoring can be detected immediately. It also means that "on the spot" analysis can be performed and feedback given to the coach and athlete immediately. The most commonly used forms of the wireless connection are:

- WiFi (IEEE 802.11a/b/g/n)
- Bluetooth

The WiFi standard refers to the wireless protocol used to transmit and receive data between the client and the server. The typical structure is where a wireless router receives data from a client and transfers the data to the server. This can be done using any number of routers or can be done using a direct connection between the client and the server without the need for any other infrastructure. Each of the protocols have different transmission rates of data depending upon the frequency of the wireless connection and antenna structure used. The transmission distance is dependent upon the antenna structure, transmission frequencies, the use of routers or boosters, and the relevant government standards governing power of transmission. The number of concurrent connections is also dependent upon the protocol used. The most commonly used of these WiFi protocols is the IEEE 802.11g and IEEE 802.11n standards. The IEEE 802.11g standard has a maximum data transmission rate of 54MBits/second and the IEEE 802.11n standard has a maximum data transmission rate of 150MBits/second.

These standards just define the connection method; they do not define the protocols used for the data transmission. Typically the standard Ethernet protocols are used over the wireless connection since the wireless connection is designed to replace the wired Ethernet connection. This means that TCP/IP and UDP are the commonly used protocols. This is the case where the difference between the TCP/IP and UDP becomes especially important. The wireless connection is more prone to transmissions errors especially when used in a mobile environment. The signal can be degraded through a number of methods including distance from the router, objects blocking the signal and other nearby wireless transmissions operating at the same frequency. As discussed earlier the UDP protocol will allow somewhat faster data transmission at the risk of losing information. The TCP/IP protocol will help prevent the loss of information at the expense of slightly slower transmissions. However, at the typical speeds of data collection, the TCP/IP slowdown will not be particularly noticeable compared to UDP but using TCP/IP will reduce the risk of information loss.

Bluetooth is another commonly used wireless method which has been especially designed for mobile devices [12]. It is designed for short range communication, low power usage, and maximum speeds around 2 – 3 Mbits/second. It allows for a maximum of 7 connections forming a “piconet” [12]. It employs a master slave paradigm with all the slave devices synchronizing to the master. For security and seamless connectivity, the Bluetooth devices need to be paired. Once paired, they can connect to each other whenever they are within range of each other without any user intervention. To operate, a driver is required on the computer and is responsible for the data transfer between the sensor unit and the computer system. The simplest method for the data transfer is where the bluetooth implements a protocol so that it appears to be a serial device but runs at much faster transfer rates that a straight serial connection. This means that the device appears like it has a serial connection and can be accessed like a serial port which simplifies the programming and access.

3. System Design

The system design encompasses the design of the entire data capturing and analysis system including the sensor data capture, the computer system, and the analysis tools. A Real time system will typically capture the sensor data in real time (wireless communication) and display the data in real time to give immediate feedback. There are a number of factors to consider when designing and implementing a system to capture the athlete data. These are as follows;

Operating System (OS). The OS is an important consideration since it needs to be able to easily communicate with the sensor unit without losing information and be easy to configure. This is especially relevant in real time monitoring since the data is streaming into the OS and it must be able to handle both the data transfer rate and store the data. The OS should be quick enough to running the analysis application or communicate with the data storage devices.

Analysis Applications. The Analysis applications are an important choice based upon the type of analysis to be performed. If the data is not real time data, then the speed of analysis is not that important. However if the data is real time from streaming sensors and needs to be displayed in real time then a fast application needs to be used that can update the display windows quickly. Interpreters or single task applications are not good for this type of application.

Concurrency. It is useful for the system to be able to perform more than one operation at a time. This enables data capture, and data analysis to run simultaneously but separately. This is the structure that is employed by most servers which need to handle streaming data.

Scalability. The system should be able to scale up to incorporate many sensors of many different types and connections. This will allow for multiple participants as well multiple sensors. This is important because it allows for versatility in the future.

4. Implementation Example

This section outlays and demonstrates the design of a real time data capturing system. The system consists of an iPhone streaming triaxial accelerometer data over WiFi and a laptop receiving the data and analyzing and displaying it as strip charts using MATLAB.

The system design consists of an iPhone 4 streaming accelerometer data to a laptop running MATLAB for analysis and display. The system consisted of a full multitasking operating system on both the sensor units (iPhones) and the laptop. Both OS'es used were more than capable to deal with communications at a capture rate of 100Hz and were also capable of concurrency but concurrency was not employed in this example. It is possible to achieve scalability by including more sensors from the iPhone or even by using multiple iPhones streaming to the laptop.

The iPhone contained triaxial +-2g accelerometers which were used to capture the motion. An application was written for the iPhone which streamed the accelerometer data over the WiFi using the UDP protocol. UDP was chosen rather than TCP/IP since the connection was a short range, reliable connection with no expected loss of packets. The network topology consisted of a direct wireless connection (ad hoc) using UDP over IEEE 802.11g. The laptop received the streaming accelerometer data and displayed it as a graph for each axis as shown in Figure 1. An application was written for the laptop using the MATLAB language to capture and display the streaming accelerometer data. This application opened a UDP socket connection, received data, and displayed the data in a strip chart format. The MATLAB application was a mixture of native MATLAB code to provide the plotting and Javascript code to provide the socket interfacing.

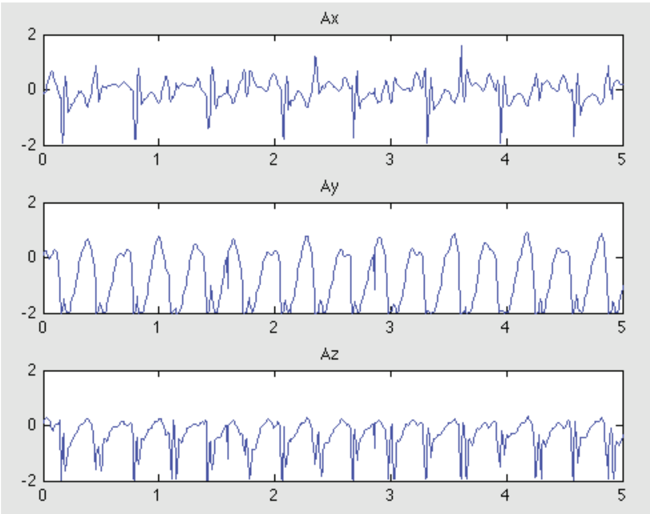

Fig. 1. A screenshot of the triaxial accelerometer data captured by the system for a 5 second window. The Ax graph corresponds to acceleration along the mediolateral axis. The Ay graph corresponds to acceleration along the vertical axis. The Az graph corresponds to acceleration along the anteroposterior axis (forward is negative). The x axis is given in seconds and the y axis is given in g's.

The system was tested by attaching an iPhone 4 to the runner at the C7 vertebrae position. The runner was told to perform a slow jog (< 14 km/hr) down a straight path. The streaming data was captured and displayed in real time as a strip chart. A screen shot of the data capture in operation is shown in Figure 1. The Ax graph corresponds to acceleration along the mediolateral axis (up is positive). The Ay graph corresponds to acceleration along the vertical axis (right is positive from the runner's perspective). The Az graph corresponds to acceleration along the anteroposterior axis (forward is negative). The footsteps and strides can be clearly seen in the Ay graph. The sidewards motion of the torso can be seen in the Ax graph and the forward surges can be seen in the Az graph. The Ay and Az axis show an unusual feature with a quick negative going spike at about 1.75s and 2.9s. This is measured from the runner and is not a communication issue since a communication issue would be seen in all axes and would appear as a zeroing of the signal and not the negative spike as shown. The reason for the spike is not known at this stage. The Ay and Az also show some saturation of the sensors at -2g. However the sensors still show the main features of the jogging. Since the data is as expected for a slow jog, then this indicates that the system was able to correctly capture the streaming data and display it in real time.

5. Conclusions

This paper has shown the important considerations when capturing data from sensors. The different methods to extract the data from the sensors were discussed. The design considerations for a sensor capture system in real time were also discussed. Finally a real time sensor capture system was designed and tested. The system was tested with a runner performing a slow jog and showed that it was possible to capture streaming accelerometer data and display it in real time on the screen of the laptop.

The ability to use smart phone to capture some running data opens up many possibilities to provide feedback to the jogger particularly when combined with the other available sensors such as GPS etc. This can take the form of data to post analyse after the session or data that can be used to provide immediate feedback to the user. This work is currently being extended by the authors through the creation of an iPhone application that will analyse the running parameters and provide feedback to the runner.

References

[1] D.Rowlands, D.James, The Application of Inertial Sensors in Elite Sports Monitoring, The Engineering of Sport 2010 (ISEA Conference of Ideas);xxxx

[2] J.G.Neville, D.A.James, D. Thiel, D.D. Rowlands, Using Inertial Sensors To Examine The Soccer Throw-in, Impact of Technology on Sport III (APCST Conference, Hawaii) 2009; 611-615

[3] D. Rowlands, D. A. James, D. Thiel, Bowler analysis in cricket using centre of mass inertial monitoring, Sports Technology 2009; vol. 2, no. 1-2: 39-42.

[4] Davey, N. and James, D.A. Signal analysis of accelerometry data using gravity based modelling, Proceedings of SPIE 2003; 5274: 362-70.

[5] D. A. James, The Application of Inertial Sensors in Elite Sports Monitoring, The Engineering of Sport 2006 (ISEA Conference of Ideas); 289-294.

[6] Herren, R., Sparti, A., Aminian, K., and Schutz, Y. The prediction of speed and incline in outdoor running in humans using accelerometry. Med Sci Sports Exerc 1999; 31: 1053-1059.

[7] Williamson, R. and Andrews, B. J. (2001). Detecting absolute human knee angle and angular velocity using accelerometers and rate gyroscopes. Med Biol Eng Computing 2001; 39: 294-302.

[8] Mayagoitia, R., Nene, A. and Veltink, P., Accelerometer and rate gyroscope measurement of kinematics: an inexpensive alternative to optical motion analysis systems, Journal of Biomechanics 2002; 35: 537-542.

[9] Blundell, B.G., Computer Hardware; 1st ed, UK: Middlesex University Press,Thompson; 2008.

[10] Blundell, B.G., Khan, N., Lasebae A., Jabbar M., Computer Systems and Networks; 1st ed, UK: Middlesex University Press,Thompson; 2007.

[11] Hewlett Packard Company, Intel Corporation, Microsoft Corporation, NEC Corporation, ST-NXP wireless, Texas instruments., Universal Serial Bus 3.0 Specification; revision 1, available from http://www.usb.org/developers/docs accessed 16 March 2011; 2008.

[12] Bluetooth Special Interest Group, Bluetooth Specification Version 4; 0, available from https://www.bluetootj.org/Technical/Specifications/adopted.htm accessed 16 March 2011; 2010.