Cloud Based Activity Monitoring System for Health and Sport

D.D. Rowlands  
Centre for Wireless Monitoring and Applications  
Nathan Campus, Griffith University  
Brisbane, Australia

L. Laakso  
Applied Physiotherapy and Exercise Science  
Gold Coast campus, Griffith University  
Gold Coast, Australia

T. McNab  
Centre for Wireless Monitoring and Applications  
Nathan Campus Griffith University  
Brisbane, Australia

Daniel A. James  
Centre for Wireless Monitoring and Applications  
Nathan Campus, Griffith University  
Brisbane, Australia

Abstract—This paper gives the concept, design, and implementation of an activity monitoring system that incorporates a database and a data analysis language as an integral part of the structure. The versatility of the design allows many different analysis techniques to be run on the extracted data. This forms the framework to allow different machine learning techniques to be applied to the data without the construction of separate dedicated systems. As an example application, this paper applies the system to determine some key features of a running based activity.

Keywords-component; activity monitoring, database, inertial sensor, health, sport

I. INTRODUCTION

Activity monitoring has long been used by health and sporting professionals to assist in their endeavours[1][2][4-8]. For health applications it represents an easy way to collect patient health information that requires patient compliance, the ability to perform cognitive tasks, and the motivation to perform the tasks required for recovery which can be hard for some patients since they will typically be tracked during a difficult period in their lives such as the postoperative period. For sporting applications activity monitoring represents a powerful way to examine the sporting activity and determine the performance metrics relevant to the sport being examined.

Inertial sensors are very useful for the determination of activity of the user [1][2] and are very low power and thus are suitable for long term monitoring. Inertial sensors are very good at determining events and have been used for performance analysis in many different sports such as cricket[4], soccer[5], and tennis[6-8]. They have also been used to determine activity in the health arena [1][2] for patients who are undergoing a specialist recovery or wellness routine. A useful platform for this has been the smart phone which commonly contain inertial sensors, radio access for data transmission and GPS for geolocation purposes [9]. Previous work by the authors have incorporated them as streaming devices into a database [9].

Typically the activity monitoring device will be a multisensor device utilising inertial sensors (triaxial accelerometers and triaxial gyroscopes) running at 100Hz to 200Hz for good resolution. Using the inertial sensors alone as the feature vector gives 6 channels without adding the timing stamp for data sequencing. For example, 6 hours continuous monitoring at 100Hz with a 7 element feature vector (time, x-y-z accelerometer, x-y-z gyroscope) will generate 6 x 60 x 60 x 7 x 100 individual data elements which is 15.12 million distinct data elements. Including other sensors in the feature vector, such as GPS, magnetometers, heart rate monitors etc will only compound the issue. The sheer volume of data generated by the available sensors means that to extract, analyse or visualize anything from the raw data will require sophisticated processing techniques [3]. These processing techniques can include simple analysis techniques through to the more complex analysis techniques including machine learning. What would be useful is a database system that is generalized enough to allow any level of analysis on the data and to allow low powered devices such as smart phones, tablets etc to access the data and perform complex analysis and machine learning techniques on the data. If the database system could perform the activity analysis then the numerical constraints would be removed from the client side allowing lighter less powerful interfaces to the data to be used. If the database incorporated a standard analysis language and was able to run the chosen scripts over the data before presentation then different analysis could be employed making the system versatile thus removing the need for a number of dedicated analysis systems.

This paper outlines the concept, design and implementation of a database system that incorporates a data analysis language (MATLAB[10]) as an integral part of the system structure. The activity monitoring with the extraction of step rate of a middle
distance runner is used to illustrate the operation of the system. This database system is the first step in producing a system that can use machine learning techniques on the analysis of activity from the multisensor data.

II. CONCEPT

This section presents the concept of a generalized database system that incorporates MATLAB as an integral part of the system structure. The idea is that the database will contain data and a repository contains the MATLAB scripts that can be applied to the data. This makes the system versatile and favours agile development since the scripts could contain any type of analysis technique and can be modified in the run-time environment. The results from scripts and the name of script executed can also be stored back into the database. This allows an audit trail so that when the modified dataset is extracted the user can know exactly what scripts and what order the scripts were applied to produce the data in its current state.

Figure 1 shows the Concept diagram of the system. The basic flow of the system is as follows: The user uses a browser to access the database, the appropriate dataset is extracted from the relational database and the selected MATLAB script is applied to the data. Any output generated by the MATLAB script, such as a plot, is displayed on the browser.

![Concept Diagram](image)

Figure 1. Concept Diagram

From the perspective of the user, the method of processing data within the system is done via a web interface. To process the data the User selects a dataset, then selects a script which contains MATLAB-formatted commands from a list of scripts and then instructs the system to start processing through a button. Figure 1 illustrates the steps in processing a dataset with MATLAB from a user's perspective using a HTTP-based client (web browser):

1. Upon the serverer receiving the HTTP request to process a selected dataset, the selected dataset is retrieved from the data access layer which is responsible for direct communication with the relational database.
2. The dataset is converted into a series of vectors by the controller and transmitted to MATLAB along with the user's selected MATLAB commands to execute against the transmitted dataset.
3. After the commands are executed, the results are fetched from the MATLAB instance and then used to create a new dataset.
4. The new dataset which is the result of the execution is inserted as a new database record via the Data Access Layer.
5. The server instructs the user that the resultant dataset can be reviewed.

III. DESIGN AND IMPLEMENTATION

The system proposed in the previous section was implemented using a variety of technologies, with a focus on extensibility so that the system may be scaled up in the future to include other features as required by the experts using it.

A. Intermediate Data Storage Format

In the monitoring phase of the performance analysis process, devices which measure the physical characteristics of the athlete output a number of different data formats as files or as a realtime data as a stream of measurements. These measurements predominately are time-based. Because the context of the measurements is to be combined with the raw data of the measurements, a custom data format was required to be implemented.

A custom XML DTD was developed for use as the internal data format for mobile sensor data. This format stores both the time-series data of the activity and also the context under which it was recorded. In this way, if the record itself was taken out of the system and transferred to another independent system or between experts, information about the measurement would remain with the data and still remain useful.

Structurally, the XML data format document has three entity tiers. Each tier can be described as:

**Context Tier**: this consists of the conditions in which the recording was taken. Examples include recording date, subject being monitored, sensor operator, device type, and unique device identifier.

**Sensor Tier**: this consists of known properties of the sensor recording a series. Examples include sample rate, range, filters used etc.

**Data Tier**: this consists of the raw data as measured by the sensor.

An example of a typical XML data format document is illustrated in Figure 2. A typical inertial sensor unit will record...
triangular linear and rotational inertial data at a given refresh rate, along with time stamps for individual samples. The root document node contains individual sensor channels and contextual information specific to the event (athlete, weather conditions, sensor device identifier). For each sensor, two subtrees will be generated: sensor properties and a sensor data container. This container is a vector-based one, and contains all the sensor’s data samples (one for each spatial dimension and one for a timestamp).

![Diagram](image.png)

Figure 2. Example XML data format document for a typical inertial sensor unit. Each of the three Entity Tiers are represented as a different level.

**B. Data Storage**

The system has been implemented using a relational database (MS-SQL 2008 R2). The XML data format documents are individual entities within the database schema. Datasets which are transformed with processing tools are given new entries in the database, and are then related to the original dataset such that any dataset can be traced back to the original, unmodified ancestor dataset. Additionally, datasets may be ‘tagged’ with keyword metadata by the owner of the data to facilitate easier searching of related datasets.

The processing tools allowed importing and exporting of datasets into the system. Flat file datasets are imported into the database by conversion to the XML data document and storage as an entry in the database using a user provided custom format mapping description. The dataset from the database can be exported in a number of formats including PDF, JSON, and CSV. The reports in PDF format are designed to be used for documentation purposes and encapsulate all the data pertaining to the selected dataset including metadata, graphs, and user annotations.

**C. Data Access Methods**

The primary method of managing data within the system is through a web-based interface implemented using ASP.NET MVC. Users are able to register their details with the site and once authenticated can navigate through the data using multiple methods, such as keyword and full-text searches. Users can quickly search for topics by clicking “topic buttons” which reveal similarly-grouped data sets.

**D. Processing**

In order to increase the efficiency of the analysis process, an element of automated data processing was required for the system. MATLAB was selected as the numerical processing tool, as it is able to run in-house developed scripts and various MATLAB toolboxes. MATLAB is interoperable with the .NET Framework via COM which allowed its integration into the system [11]. Any plots generated were plotted on the client (browser) using the jqPlot plugin[12].

**E. Web Interface**

An example of the implementation of the website interface can be seen in figure 3. The interface consists of two sets of tabs across the top of the page with the associated display area for the current tab below that.

![Image](image.png)

Figure 3. The web interface design showing the two tab bars and the viewing area underneath them.

The first row of tabs is the system tab bar for the general use of the system. The “Home” tab allows the user to read the home page giving information about the database system. The “Data” tab allows the user the choice to choose a dataset from a list of known datasets or to upload a new dataset into the database. The metadata can also be added. The “Community” tab is unimplemented but it is envisaged that it will contain discussion boards etc. The “Support” tab is the help for the system. The “Log Off” tab allows the user to log out of the system.

The second row of tabs is the analysis tab bar for the analysis of the data. The “Overview” tab gives information about the specific dataset being viewed including the metadata and allows for the export of the data. The “Accelerometry” tab plots the accelerometer data with the option to view the plot in full screen mode. The user can select a region to zoom into in either of the two graphing views. The “Map” tag displays the GPS data on a map showing the location of the activity. The “Process” tab allows the editing of the MATLAB commands to be run over the data. The Discussion tab allows metadata describing the graph to be added to the dataset.
IV. ACTIVITY MONITORING EXAMPLE

Inertial sensors are ideal for monitoring activity. A simple activity measure that encompasses both the Health and the Sports arena is the step rate. In Health, this can be used to give an indication of the intensity of the exercise so that it can be strictly controlled by the health professional during rehabilitation. In running based sport applications this can be used as an indicator of the training intensity, which can be useful to the athlete and the coach. The system was applied to monitoring the running activity where a subject was asked to run one lap around an oval. The aim was to collect the data and report on the activity by running a script to plot the data and another script to calculate the step rate over the duration of the activity.

The system was deployed onto a laptop and a smartphone was used to capture the inertial data for the activity and stream it back to the laptop once the authentication into the system had been completed. The subject was fitted with a harness on the thoracic part of the spine which held the smartphone firmly in position. The smartphone was preloaded with a purpose written monitoring application, which streamed triaxial accelerometer, 3 axis gyroscope, latitude, longitude and heading data to the laptop at a user-defined rate across a TCP/UDP connection which then stored it as a file for later analysis. Ten trials were conducted where the subject was instructed to run at a moderate pace once around the track until the subject reached the initial starting point.

At the conclusion of each trial, the trial metadata was modified to reflect the circumstances of the trial. Also at the conclusion of each trial, two MATLAB scripts were executed over the captured data according to the method outlined in section II, with the results displayed to the subject. This gave the subject feedback upon the previous session.

The first MATLAB script was executed to plot the triaxial accelerometer data for a portion of a single trial. Figure 4 shows a plot of the triaxial accelerometer data at a 30Hz sample rate as plotted by the system and displayed via the web interface. In this figure, the steps can be easily seen in the inertial sensor Z axis with the forward surges and the sideways rocking motion due to changing footsteps in the stride visible in the other inertial sensor axes. Other features such as toe-on and toe-off can be seen in the steps given in the inertial sensor Z axis. The consistent step rate can also been seen in the plot.

The second MATLAB script was executed to determine the step rate. This script was run over a single trial and counted the number of peaks in the trial and determined the average step rate. For the example given in figure 5, the average step rate was found to be 2.55 steps/second. Figure 5 shows the step rate per second for every second of the trial. It can be seen that the step rate varies between 2 and 3 steps/second.

V. DISCUSSION AND CONCLUSIONS

This paper has detailed the concept, design and implementation of a database system incorporating the MATLAB data analysis language as an integral part of the system structure. One advantage of this method is that the numerically intensive and hence power hungry operations are performed on the server allowing mobile devices to be able to perform the analysis on data over the web and save/visualize the results. It should be noted that the response time of this system will be dependent upon a number of factors such as the complexity and size of the analysis code, the amount of data that needs to be processed by the analysis code, the processing power of the server, the load on the server and the network connection and network load. The activity monitoring example given in section IV required less than 5 seconds for a remote offsite connection connecting through a VPN.

Another advantage of the system is the versatility that it affords. Since the numerical analysis is run using scripts then this implies that many different analysis techniques can be run on the extracted data. This forms the framework to allow different machine learning techniques to be applied to the data without the construction of separate dedicated systems for each new technique. The user only has to build the technique into a script. A further advantage is that the system allows the user to take advantage of the toolboxes that form a powerful part of the MATLAB environment. A further advantage is that the
metadata is used to store auditing information about the processing of the data (ie. Scripts that were run) with the new results stored as a new entity in the database. This method allows have access to the historical data as well as understand the history of preprocessing that gave the data in its current form.

This paper applied the system to determine some the key parameters of an activity being monitored. The activity was running and inertial sensors were used to monitor the performance. Two MATLAB scripts were run over the data, one to plot the data, and the other to determine the step rate. The plot allowed the cadence to be seen and the allowed key events such as footfall, toe-on and toe-off to be seen. It also allowed the forward surging to be seen as well as the rocking motion of the torso due to the runners striding pattern. The step rate gave an indication of the speed of the runner. The fact that key events, and trends can be successfully extracted from the running activity means that these can be used as the inputs to machine learning algorithms for the activity monitoring.

Currently the database system is a proof of concept prototype and is not particularly strong in the security of the transfer of the website or data. Future work will involve improving this by the incorporation of the http secure protocol to bring it a level suitable for general use.

Overall, this paper has shown that a versatile system has been created which can form the basic platform for machine learning algorithm design.

REFERENCES


