Situated Display in Hospital Ward

Ivan Naumovski
ITU
Rued Langaardsvej 7, 2300
Copenhagen
inau@itu.dk

Martino Secchi
ITU
Rued Langaardsvej 7, 2300
Copenhagen
msec@itu.dk

Diem Hoang Nguyen
ITU
Rued Langaardsvej 7, 2300
Copenhagen
hong@itu.dk

ABSTRACT

Author Keywords
Guides; instructions; author's kit; conference publications; keywords should be separated by a semi-colon. Mandatory section to be included in your final version.

ACM Classification Keywords
H.5.2. Information Interfaces and Presentation: Input devices and strategies; I.2.6. Artificial Intelligence: Learning

See: `http://www.acm.org/about/class/1998/` for more information and the full list of ACM classifiers and descriptors.

INTRODUCTION
Since the dawn of the computer age modalities have been an important aspect to consider when interfacing with the machines, whether it has been done using keystokes, audio-control or gestures.

It has always been of high relevance to push the limits for how we interact with technology, mainly to empower the individual.

Technology is supposed to make common tasks even easier to achieve. Consider the case of a physically disabled person being empowered by using an alternative modality such as voice control.

In our case the focus is to improve on how people in the healthcare sector interact with technology. In the healthcare sector interaction can actually prove to have negative effects, particularly bacteria can be transferred. For example, in most operating room at hospital, large displays and computers capture medical information during surgery. Because of sterility, the input devices are not handled directly by surgeon but by assistants or nurses who may cause some misunderstandings and delays[1]. The wearable devices are suggested to resolve this problems.

One way to avoid spreading bacteria is by avoiding physical interaction with input devices.

This can result in less time spent scrubbing or rinsing as the likelihood for microbacteria being transfered via input devices is reduced.

Ultimately alternative modalities could provide employees within the sector a way to use native hand gestures to perform tasks such as interacting with x-ray images.

Background There exist alot of prior work within the field of HCI which is relevant to our solution. The papers which influenced this project the most will be mentioned in following section.

Alternative Modalities
Prior work in regards to pattern recognition has been of high relevance to our solution. bla bla bla bla

Smart Machines
Prior work in regards to pattern recognition has been of high relevance to our solution. bla bla bla bla

SYSTEM
The system we have designed consists of multiple parts, one being a input device, another being a data processer, a third one being a communications service and the last one being a presenter.

The Input Device
Our motion tracking device(MTD) has been built in such a fashion that it resembles a wrist watch. This form factor makes it rather compact. Additionally a lot of people wear watches on a daily basis which makes the device of a recognizable morphology and barely noticeable for people accustomed to such devices. The MTD contains a list of components, the most important are 6DoF Sensor, Bluetooth and the Arduino microcontroller.

The **6DoF Sensor** is the heart of the device. It is a board which contains an accelerometer and a gyroscope. This means that it is possible to measure movement and rotation of the wrist of anyone wearing the device. The sensor produces a set of 6 values: 3 for acceleration and 3 for rotation, each representing information on the axes of the coordinate system. The input received from the 6DoF sensor is then propagated to the **arduino microcontroller**. The arduino prepares the raw values into JSON data. The JSON is then sent using the bluetooth device.

Weka Gesture Recognition System
Since the input device we have built is quite lightweight, data processing needs to be done elsewhere. This provided us with

Figure 1. The input device.

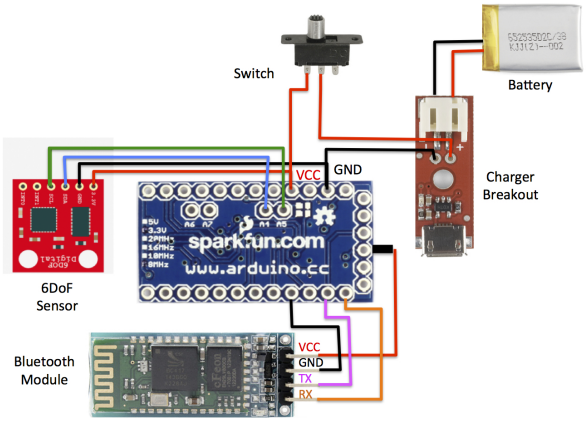

Figure 2. This schematic shows the circuitry on the device.

the challenge of transferring data from one bluetooth capable device to another. All the processing and preprocessing of the device's data is performed on a desktop application on a computer connected to the device. Especially challenging was to find and use Java libraries that would allow to establish a Bluetooth connection with the device and to exchange data with the desktop application that would process it. For further technical details about the challenges encountered please refer to the System implementation section. This component of the system is responsible for the preprocessing of the data and the gesture recognition. Every time a gesture is captured by the system, an identifier of the gesture is sent from the desktop application to the webservice, where it will be available for its final consumer: the android application.

Android Application

The android system is a quite limited system. It is able to display an image. It provides the user with the possibility of panning and zooming a given image. When the user zooms all the way out panning provides additional functionality. Pannning left and right will then swap images from the set of loaded images. add some user scenarios - images etc.

Webservice

The communication between the Weka Gesture Recognition System and the Android Application is done by a simple webservice. It provides GET, POST and DELETE requests which manipulate a queue.

GET pops all the queued gesture recognitions, POST pushes a new one to the service and DELETE clears the queue.

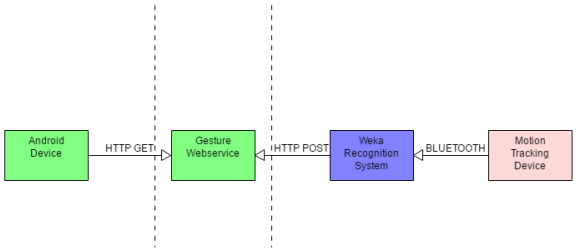

Figure 3. This schematic shows the system interactions.

SYSTEM IMPLEMENTATION

hue hue hue

Gesture Recognition

Preprocessing

Once the acceleration and rotation data is sent from the device to the computer, the values are smoothed with an average of the 20 previous values in order to avoid and reduce the effect of noise on the sensors. This phase is called preprocessing of the data. This allowed us to have more precise information, as can be deducted from the images 4 and 5

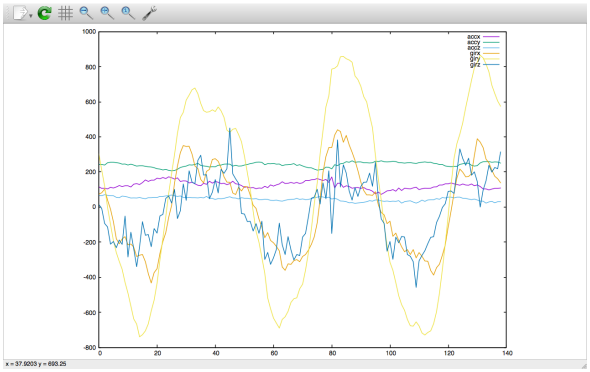

Figure 4. Data from the device before preprocessing.

Evaluation

For the actual processing and gesture recognition, we use a sliding window approach on the time series data we receive. The window is of size 50, hence it will evaluate a sequence of 50 movement instances in a sequence, every instance consisting of 6 values. The window is able to capture a period of time of about 2 seconds, as the sampling rate of the device is about 25 Hz. We organized known gestures in a very large training set, each individual gesture stored as a list of 50 * 6 values plus an identifier, in a way that every gesture will be compatible with the sliding window. The training data set in the final version of the project reached more than 500 entries

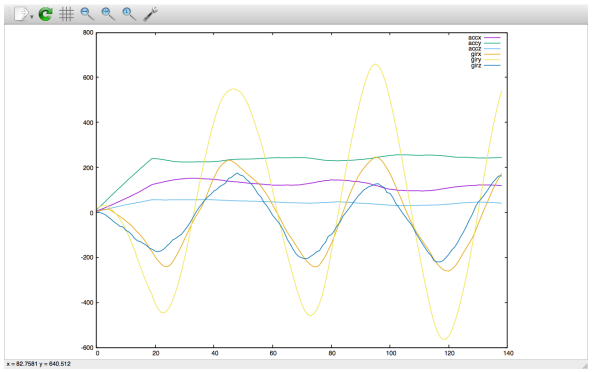

Figure 5. Data from the device after preprocessing.

in total. We use Weka 3.6 to evaluate newly received data using a BayesNet classifier and comparing it to the training set. This resulted as the most accurate classifier for our model, giving up to 100% accuracy when testing it. The evaluation of the gesture is performed every 10 * 6 new values received.

Technical challenges

maybe move this part to discussion? One of the hardest challenges encountered during the implementation of the system was the interaction with bluetooth and the desktop application for gesture recognition. The application was programmed in Java, so that it was possible to use the Weka libraries or the gesture evaluation. We used the BlueCove Java library to build a bluetooth interface [**?**] cite bluecove http://bluecove.org/ in lack of a better alternative. This library has no longer been updated after 2008, and has some compatibility issues with modern operating systems and java versions. In particular, OS X deprecated functions that were required for some of the library functionalities after the release of OS X 10.8 in 2012. Also the library is bound to work in 32 bit mode, which is not supported for java sdk 1.8 nor 1.7. The only way to make it work was then to downgrade the application to use Java 6.

RESULT

DISCUSSION

One of the less performant aspects of the project was the bluetooth connection, which was very unstable. We account interference in the ITU building mostly responsible for that. other reasons?

ACKNOWLEDGMENTS

We thank CHI, PDC and CSCW volunteers, and all publications support and staff, who wrote and provided helpful comments on previous versions of this document. Some of the references cited in this paper are included for illustrative purposes only. **Don't forget to acknowledge funding sources as well**, so you don't wind up having to correct it later.

REFERENCES

1. Jalaliniya, S., and Pederson, T.*Designing Wearable Personal Assistants for Surgeons: An Egocentric Approach*. IEEE Pervasive Computing Magazine 2015.