

Cloud Based Patient Monitoring Platform Using Android Smartphone Sensors

Vlatko Nikolovski, Petre Lameski, Ivan Chorbev

*Faculty of Computer Science and Engineering, University of Ss Cyril and Methodius, "Rugjer Boshkovikj" 16, PO Box 393, 1000 Skopje, R. of Macedonia
Emails: vlatko.nikolovski@finki.ukim.mk petre.lameski@finki.ukim.mk ivan.chorbev@finki.ukim.mk*

Abstract: *This paper presents a proof of the concept cloud based patient monitoring and self-care platform, powered by measurements provided from various smartphone sensors. The Cloud platform provides the infrastructure and computational capacity for calculation of the navigation and motion tracking system, fall detection monitoring, as well as emergency notifications. The navigation system uses the pedometer and fusion of the accelerometer, gyroscope and magnetometer sensors. It aims to estimate precisely the patient's movement and location. While both navigation and tracking systems can independently determine the incremental movement and indoor localization of the patients, they are fused in order to provide more accurate estimations. The fall detection monitoring is enabled by processing the raw data collected from the smartphone's accelerometer and gyroscope. Furthermore, the cloud system provides various statistics for the physical activity of the patients, based on measurements from the pedometer. Consequently, this paper proposes a proof of the concept cloud based platform that is scalable and highly responsive, used for real-time monitoring and tracking a large number of patients. It also provides indoor navigation and other self-care features.*

Keywords: *Self-care, home care, patient monitoring platform, android sensor fusion, step counter, cloud computing, indoor navigation.*

1. Introduction

The emerging ICT technologies, used to develop better healthcare platforms for patients, are still explored. Despite the increase of the researches for improvement of the way we provide self-care and emergency care, many challenges remain in designing more sophisticated, reliable environment. During time-critical health emergency situations, the ability to automate the tasks of monitoring and tracking patient's condition is a key benefit for increasing the quality and quantity of patient care. In such environments, where the time is of importance, there is little tolerance for errors and unreliability.

We propose a cloud based Core System Architecture built on top of the Windows Azure cloud platform [1]. In addition, the cloud based architecture [2] is closely tied with the mobile platform architecture providing scalable platform services (SaaS, PaaS and IaaS). Furthermore, these functionalities promote platform independence, simplify the interconnectivity and interoperability and guarantee high-availability of the services. These key benefits promote the cloud architecture as the most suitable for building a real-time high availability platform. Our goal is to provide a reliable cloud platform for monitoring numerous patients through the following technologies:

- Sensor measurements provided by the android mobile devices and the Kinect sensor. Each mobile device carried by the patients, exports its sensor data to the cloud Core System. The sensors exported to the cloud are: accelerometer, gyroscope, magnetometer and camera [10]. The data obtained from a deployed Kinect sensor on site is also included in the cloud Core System.
- Cloud based Core System. Following the cloud architecture principles, the system is scalable, highly-available with virtually limitless computational power. By processing the sensor's data with different algorithms, the system provides several functionalities, such as: patient navigation and tracking, fall detection monitoring, emergency notifications, as well as providing statistics for the physical condition of the patients.
- Secure user-friendly web interface, allowing authenticated users to preview and share real-time patient information, as well as various patient metrics and statistics, based on the accumulated data for each patient.

2. Android sensors and sensor fusion algorithms

A lot of researches have been done so far in the field of mobile sensors, their physical characteristics, techniques for noise reduction and sensor fusion techniques. In addition, the primary data feed in the Core System originates from the mobile sensors, so that the system must implement algorithms for calibration, noise reduction and sensor fusion in order to provide reliable and accurate functionalities.

The Android sensor framework uses a standard 3-axis coordinate system to present data values. The coordinate system is defined relative to the device's screen for most of the sensors. In addition, when the device is in its default orientation, the X axis lays the horizontal pointing to the right, the Y axis is vertically pointing up,

and the Z axis points usually to the XY coordinate plane. This coordinate system is applicable among all three hardware-based sensors: the acceleration sensor, gyroscope and magnetic field sensor. Each sensor accumulates errors in the output resulting from the noise of the signal, the surrounding environment and stemming from the physical characteristics of the material that the sensors are made of. Thus, the raw data from each of the sensors must be calibrated to remove the noise, before it can be fused with other sensor's data. Different methods and techniques are used by applying Kalman filter, particle filters and complementary filters [5].

However, the process of filtering and calibration is not trivial, because it is loosely tied to the dynamics of the system. In addition, if the system converges from one state to another very fast, the Kalman filter technique gives far better results than the complementary filters. Likewise, the complementary filters provide better filtering and noise reduction when the system changes slowly the states. Assuming that the patients move very slowly, we intended to use complementary filters because the changing rate of the system is very low. Fig.1 shows the output data from the android sensors in uncalibrated state, as well as in the calibrated state. The plotted data was obtained from the mobile device in a manner of moving the device at 90 degrees up and then returning it to the starting point. The filtered data is far more accurate than the raw calibrated data providing better estimations later in the process of fusing the sensors.

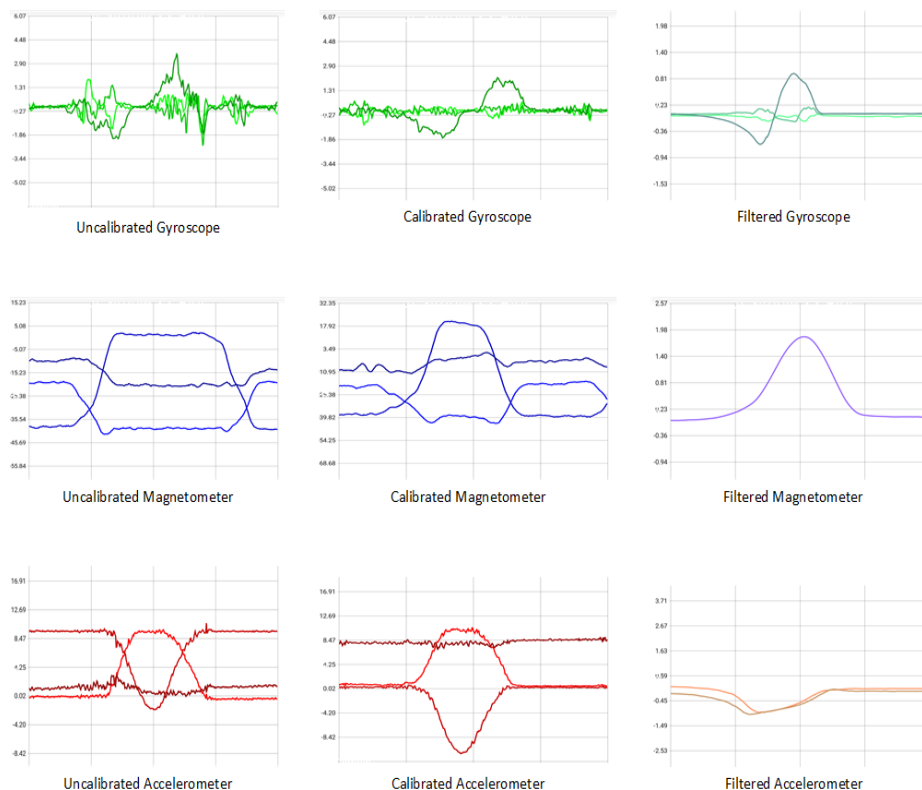


Fig. 1. Raw output from android sensors

2.1. Navigation

In order to provide precise estimation of the device's physical position into the world frame of reference, two components need to be provided: the rotation and translation values of the device. The rotation vector is represented by the Roll, Pitch and Yaw vertices' values.

There are various different functionalities so far that are implemented over the fused mobile sensors. As presented in [6], a complementary filter is used to calibrate the output offset and estimate the positioning of the device. Only short time intervals are taken in consideration to avoid the gyroscope drift. In addition, low-pass filtering is applied over the accelerometer and magnetometer, and high-pass filtering over the gyroscope signal to calibrate the drift and bias errors. With this fusion technique, the system calculates the dimensions of Roll, Pitch and Yaw of the device that are representing the rotation of the device. The author in [11] implements another methodology, using a Kalman filter to suppress the gyroscope drift and bias error from other sensors. Furthermore, an algorithm is proposed to deal with the calibration and output offset errors.

Our proposed solution that estimates the rotation vectors uses estimations from all three android sensors, the accelerometer, gyroscope and magnetometer. Fig. 2 shows the implemented technique of fusing sensors.

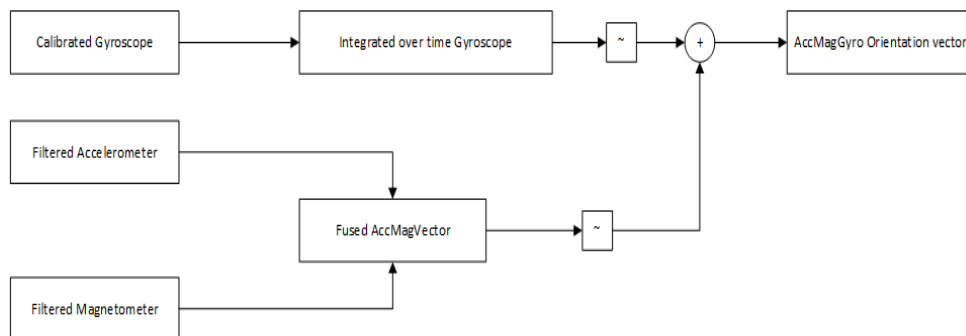


Fig. 2. Sensor fusion technique

The fusing accelerometer and magnetometer sensors can provide an estimation of the rotation of the devices that is very noisy. The noise relates to the magnetometer sensor which includes a lot of noise. In addition, the accelerometer provides the gravity vector and the magnetometer provides the compass direction. In this aspect, the gyroscope is far more accurate and has a short response time. The problem with this sensor is the gyro drift. In order to calculate the rotation for all three axes, the output data must be integrated over time. During this process, small incremental errors are introduced at each iteration. These small errors yield the gyro drift phenomena. To suppress both the noise orientation and drift, we apply gyro changes only for changes in short time intervals, while using acc/mag data for long periods of time. Fig. 3 shows the plotted data output from the two presented techniques that are later fused into one output that tends to suppress the noise in sense of better estimation of the rotation vectors.

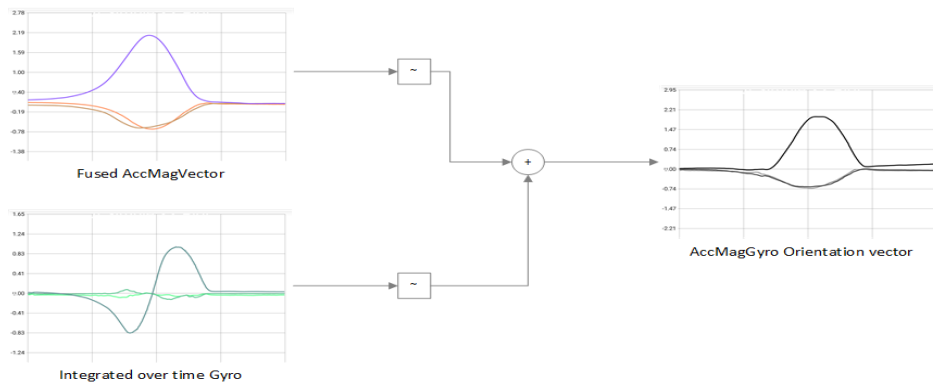


Fig. 3. Sensor fusion raw output

2.2. Pedometer

Another great functionality extending over the android sensors is the step-counting methodology. Based on the calibrated 3-axis accelerometer, the authors in [7] propose an algorithm for efficient step-detection pedometer. The most important feature of this pedometer algorithm is that the method is efficiently applicable to any variable speed of walking, presenting a dynamic threshold that varies over time. Various implementations of the step counter methodology are known so far, such as accelerometer based calculations, gyroscope based calculations, as well as combination of the two sensors in order to provide a better step detection. Fig. 4 presents the raw data output from the accelerometer and gyroscope sensors, under normal conditions of walking, as well as balance disorder walking. In addition, we can extract a pattern of recognition that successfully defines the walking disabilities of the patients and the count steps taken.

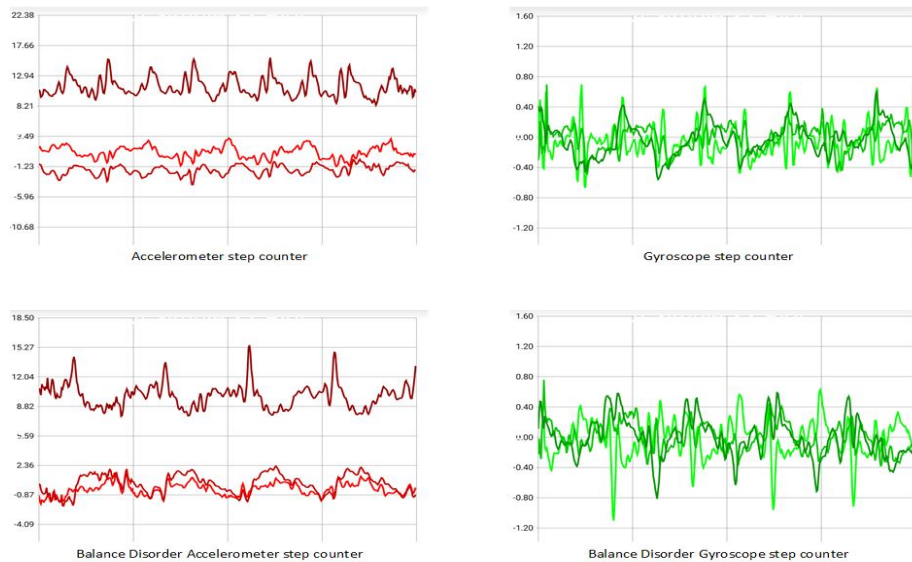


Fig. 4. Accelerometer and gyroscope step counter output

Both rotation and pedometer data are fused to estimate the patient position and provide navigation and tracking. The rotation vector tracks the orientation of the patients while the step counter detects if the user is moving.

2.3. Fall detection

Fall detection of the patients is another functionality provided by processing the sensor's data. As presented in [9], the algorithm uses a 3-axis accelerometer and gyroscope in order to determine the threshold value of the falls. The algorithm recognizes four different states of the human body. During every transition between different states, the data from the accelerometer and gyroscope is processed to determine if a fall event has occurred. Additionally, to verify the detected fall, we use the deployed Kinect sensor. We use the changes in the average distance from the Kinect sensor as an input. The goal is to detect sudden movements that are unusual for the users that are confined to bed or in a wheelchair. Based on our initial data from the usage, there is a large difference in the data from the Kinect when the user moves quickly or falls. This can be observed in Fig. 5 – left.

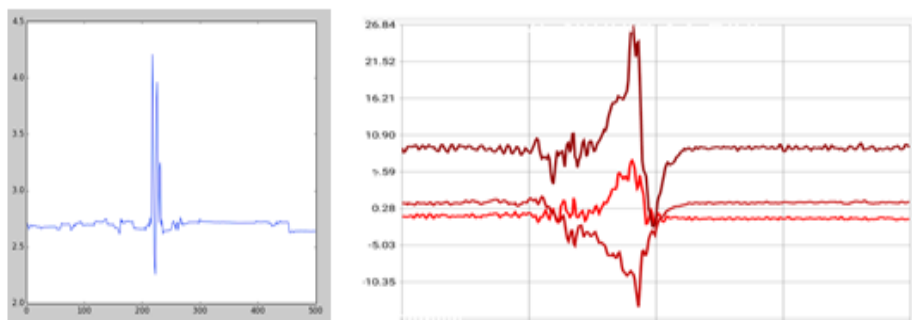


Fig. 5. Fall detected by the accelerometer and the Kinect sensor measured separately by their corresponding modules (left – Kinect measurement, right – accelerometer measurement)

The data is obtained from a healthy person that makes sudden movements. It can be observed that the normal movement of the person while sitting differs drastically from the situation when the person quickly stands up or falls. Using simple thresholding, we are able to distinguish the normal behaviour from falls and sudden movements similar to [12] with the difference that we only take into considerations of the movements of the upper body parts (head and shoulders). The data obtained from the Kinect is sent to the Core System that processes and keeps history of the patients' data. This data is also combined with the accelerometer from the mobile device and if both detect anomaly in the behaviour, a red alarm is turned on and the doctor is notified that the patient needs urgent attention. The Kinect sensor gives a good addition to the mobile accelerometer giving a validation of the findings of the mobile accelerometer. The mobile accelerometer fall detected is given in Fig. 5 – right. As it can be observed in Fig. 5, it is relatively easy to detect the fall as a large change in the distance from the Kinect or a large value in the acceleration detected by the accelerometer.

3. Cloud based framework

The cloud based core framework is developed on top of the Azure cloud infrastructure. The proposed model of the core framework tends to provide scalable and flexible infrastructure for building a heavy load environment. The core framework provides a communication protocol for connection with various Android mobile devices. The protocol is based on the pure HTTP [3] and Web Socket [4] protocol that have built-in features for two-way communication between the Core System and the devices attached to it. Fig. 6 shows the primary architecture of the model that represents the core cloud framework.

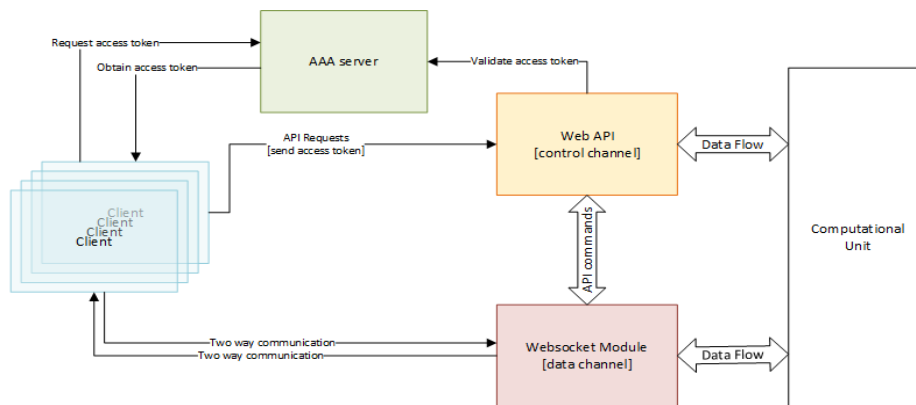


Fig. 6. Model of the cloud based core system

The model is capable of handling third party services that need to integrate with the platform and can support various platform independent clients. The ad-hoc service integration and platform independence relies on the simple communication protocol of the model based on pure HTTP and Web Sockets. Loosely-coupled environment is the main characteristic of the model, allowing every component to be easily replaced without breaking the whole system.

The workflow presented by the model tends to differentiate two communication channels, a control channel and a data channel. The control channel is implemented as a standard web API, while the data channel is based on the Websocket protocol.

The AAA (Authentication/Authorization/Accounting) component of the model provides authorization through OAuth2 [14] standard. Each authorized client is provided with an access token created from the AAA server. The access token is used in every client request in order to authenticate with the platform. According to the Websocket protocol defined in [13], the Websocket security model uses web browser origin to restrict the access to the Websocket server. When the client is not a web browser, the origin based security is not applicable. Because the Websocket security model cannot provide a reliable security out of the box, the model implements a work-around. When a client initiates a connection with the core system, the system provides the client with an access token. When the client sends control data messages to the system, it uses the control channel that is secured by the Web API standards.

On the other hand, when the user exports sensor's data to the system it uses the data channel based on Web Sockets. In order to provide a reliable security model, the

custom security model consists of integrating the access token into every data message that is sent to the system. Furthermore, the Websocket module validates the token received from the client against the AAA server. In this way we are providing a reliable security model for the Websocket module.

The Web API module is just an interface for interconnection with various services, while the Websocket module provides two way communication between the clients that sends sensor data to the cloud platform for further analysis. Both the Web API and Websocket modules are transmitting control and data messages to the Computational Unit.

The Computational Unit is a module that implements several algorithms for sensor fusion and provides calculations based on the sensor's data that is transmitted to this unit.

Fig. 7 shows the current software implementation of the proposed cloud infrastructure. On top of the infrastructure there is a load balancer that balances the traffic among each endpoint. The balancer differentiates the HTTP traffic from the web-socket TCP traffic, providing two independent channels of communication, the control channel and the Websocket channel.

The AAA server provides authorization and accounting functionalities, as well as handles the validation of the access tokens.

On behalf of the load balancer capabilities of balancing and differentiating the traffic, the system consists of multiple Websocket servers and multiple web API servers. That infrastructure provides a high load environment that is capable of serving millions of concurrent client sessions.

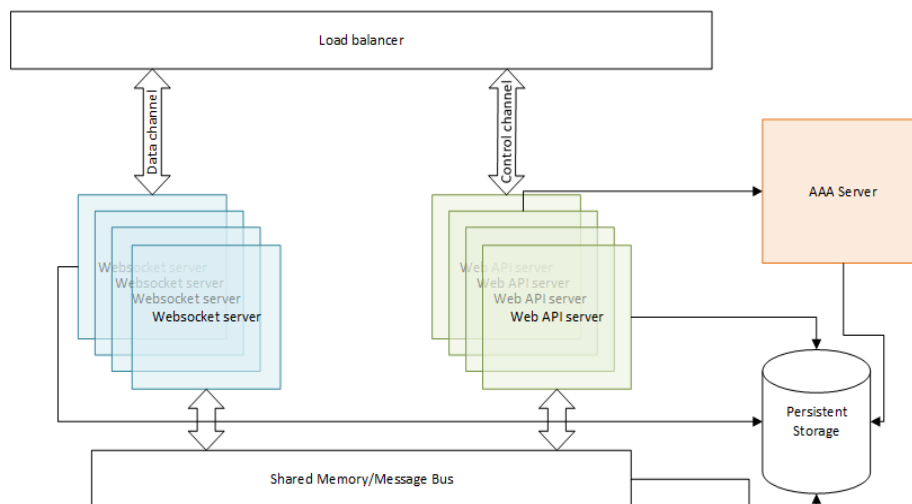


Fig. 7. Software implementation model of the cloud based core system

The shared bus layer provides session's persistence over the servers. In addition, each server saves the session's data to the shared bus, so that the user can easily be shifted between the servers in a disaster recovery scenario. Also the bus provides a functionality of shared memory between the servers for interposes communication.

4. Methodology

Each device connected with the Core Cloud System sends raw data from its sensors, such as: accelerometer, gyroscope, magnetometer and camera [10]. In addition, due to the high computational power of the Cloud, the intensive computations over the raw data are processed only by the Core System. Consequently, the framework spares the CPU and reduces battery consumption in the mobile devices. Based on different fusion techniques and algorithms for processing sensor data, the patient platform provides several functionalities, such as: navigation and tracking of patients, fall detection monitoring, emergency notifications, as well as providing statistics for the physical condition of the patients.

The navigation functionality provided by the Core System consists of sub-functionalities. The main characteristics for determining the device physical position in the world frame of reference are the rotation and translation values of the device. In addition, the first functionality is provided from the algorithms for navigation, which calculates the orientation of the patient. The second sub-function is provided from the step-counting functionality. The system fuses the data provided from the rotation and the pedometer components in order to estimate the patient's position presented in local coordinates. The quality of the navigation service under such conditions is very poor, in lack of precise mapping of the local position of the patient to the world frame of reference.

The task of navigation however is not trivial, so that in order to translate the local co-ordinates of the patient to global coordinates multiple reference points are needed. Each reference point contains global coordinates and when a patient's device checks at a specific point, his position is mapped onto the global reference network. The reference points are predefined for each room and are inserted in the Core System. The localization of the patients is done by finding the nearest reference point to the currently detected location of the patient. The trajectory of movement is interpolated from the reference points, on which the patient's device has checked. This process is illustrated in Fig. 8.

Furthermore, the Core System monitors falls of the patient through a fall detection module. When fall detection occurs, the alarm is triggered by the Core System with the estimated position of the patient. To improve the reliability of the fall detection system, we are using both the accelerometer data and the Kinect data to estimate the fall. Both of these functions are independent detectors and each of them estimates whether a patient movement is a fall. If both modules return a signal that the patient has fallen, the nearest doctor is immediately informed. Both modules log the data for each patient and the system analyses the correlation between the two signals offline. By using the correlation between the data obtained from the Kinect sensor and the data obtained from the accelerometer, we plan to be able to predict false detections caused from the sensors and more importantly, false negative predictions where there is a fall, but the system fails to detect it.

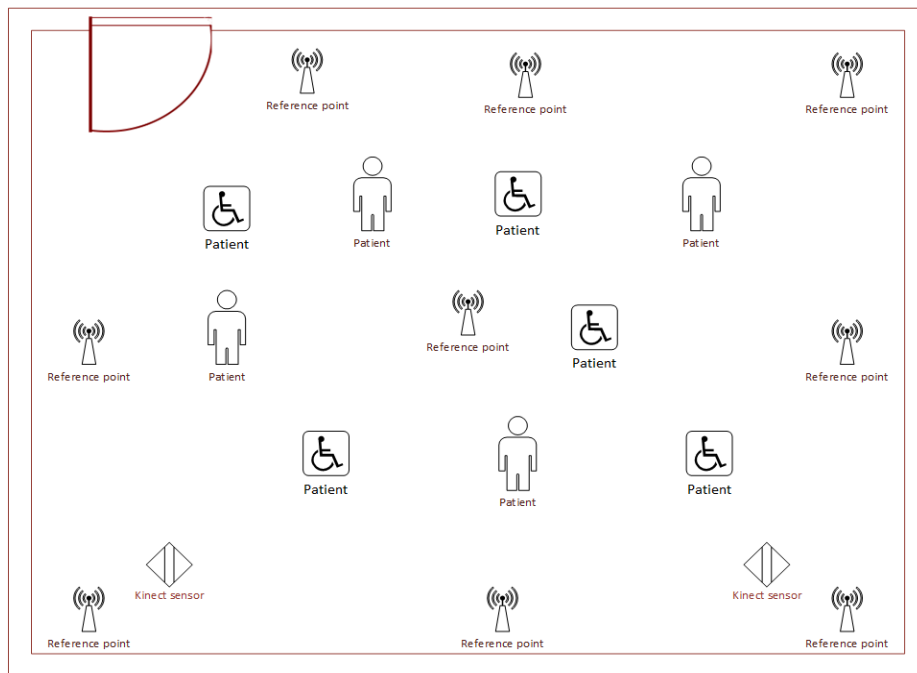


Fig. 8. Proof of the concept sensor network infrastructure for monitoring patients

Based on the data collected from different sensors and implemented algorithms, the Core System provides different metrics and statistics for every patient, providing personalized profiles. By providing the detected reference points at certain periods of time, we are able to record the patient's periodic behaviours and activities. After a longer period of surveillance, the system will be able to establish the typical patient hourly, daily and weekly behaviour that would help in the treatment of some diseases, especially mental disorders, where the detection of specific patient's patterns might be crucial in determination of the disorder and treatment.

The sole process of generating the user model or profile will allow the system to detect anomalies in the patient's behaviour on a larger scale and not only anomalies in the current movements of the patient, like changes in the living habits or unusual behaviour during certain parts of the day, which could give notifications of certain improvements or degradation of the patient's health.

5. Conclusion

Using the pedometer component, the system gives information about the physical activity, counting how many steps the patient has made. Furthermore, based on the fall detection component, the system aims to find a related pattern of interconnection between the falls of the patients. The patient's behaviour surveillance that is provided by the system will allow detection and early prevention of the consequences of falls or other types of health risks in patients, especially in elderly patients.

The system proposed in this paper is easy to deploy in any environment, since it relies on relatively cheap and available mobile devices and in addition, on the easily

available Kinect sensor that will provide reliable data about the patient. It is also power/resource efficient because all the processing are done in the cloud. All data from the sensors for each patient is logged separately.

Eventually, the cloud aggregation of the sensory data, provides rapidly increasing datasets of raw data that will soon be subject of thorough data mining and additional research. Analysis of the data about the human habits, behaviour, changes in movement will generate models for diagnostics, automated self-care and home care. Smart homes connected to the cloud will provide reliable and certainly more affordable healthcare for those in need, especially for the aging population in Europe.

References

1. Redkar, T., T. Guidici. Windows Azure Platform. New York, Apress, 2009, pp. 53-104.
2. Fielding, R., J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee. Hypertext Transfer Protocol-HTTP/1.1, No RFC 2616, 1999.
3. Fette, I., A. Melnikov. The WebSocket Protocol. 2011.
4. Lin, C. F. Modern Navigation, Guidance, and Control Processing. Vol. 2. Englewood Cliffs, Prentice Hall, New Jersey, 1991.
5. Ayub, S., A. Bahraminisaab, B. Honary. A Sensor Fusion Method for Smart Phone Orientation Estimation. – In: Proc. of 13th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting, Liverpool, June 2012.
6. Sheu, J. S., W. C. Jheng, C. H. Hsiao. Implementation of a Three-Dimensional Pedometer Automatic Accumulating Walking or Jogging Motions in Arbitrary Placement. – International Journal of Antennas and Propagation, 2014.
7. Weng, S. K., C. M. Kuo, S. K. Tu. Video Object Tracking Using Adaptive Kalman Filter. – Journal of Visual Communication and Image Representation, Vol. 17, 2006, No 6, pp. 1190-1208.
8. Li, Q., J. Stankovic, M. Hanson, A. T. Barth, J. Lach, G. Zhou. Accurate, Fast Fall Detection Using Gyroscopes and Accelerometer-Derived Posture Information. – In: Proc. of 6th IEEE International Workshop on Wearable and Implantable Body Sensor Networks, 2009. BSN'2009, June 2009, pp. 138-143.
9. Android Sensor Support Platform for MATLAB.
<http://www.mathworks.com/hardware-support/android-sensor.html>
10. Abyarjoo, F., A. Barreto, J. Cofino, F. R. Ortega. Implementing a Sensor Fusion Algorithm for 3D Orientation Detection with Inertial/Magnetic Sensors.
11. Bevilacqua, V., N. Nuzzolese, D. Barone, M. Pantaleo, M. Suma, D. D'Ambrosio, A. Volpe, C. Loconsole, F. Stroppa. Fall Detection in Indoor Environment with Ki-nect Sensor. – In: Proc. of 2014 IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA), 23-25 June 2014, pp. 319-324. doi: 10.1109/INISTA.2014.6873638.
12. Fette, I., A. Melnikov. WebSocket Protocol. Internet Engineering Task Force (IETF), December 2011.
13. D. Hardt, Ed. Microsoft. The OAuth 2.0 Authorization Framework., Internet Engineering Task Force (IETF), October 2012.