IS-ACTIVE

Inertial Sensing System for Advanced Chronic Condition Monitoring and Risk Prevention

WP4 – Algorithms

Document ID:	D4.1
Document title:	Distributed feature extraction
Document type:	Prototype
Contributors:	INE, UT

Table of contents

1. Overview.. 3
2. Feature extraction methods.. 3
2.1. Activity level - the IMA value...3
2.2. Exercising - detection of the direction of vertical motion.....................................4
2.3. Extracting features for automatic recognition of object handling 6
3. Conclusions 8
4. References 8

1. Overview

This document briefly overviews the feature extraction mechanisms developed in the IS-ACTIVE project. The results are available as prototypes, used for experimentation and evaluation.

2. Feature extraction methods

The project requirements indicate two distinct usage scenarios for the sensors: during daily activities (continuous monitoring) and during physical exercising (temporary monitoring and assistance). Starting from these scenarios, the following feature extraction methods have been developed as building blocks for the end-user applications.

Firstly, the estimation of daily activity level is implemented through the IMA value, corresponding to the body energy expenditure. This feature classifies under continuous, long-term monitoring. Secondly, physical exercising is augmented with automatic, sensor-based detection of vertical movement for a (dumbbell) lifting exercise. This feature classifies therefore under temporary, short-term monitoring and assistance. Finally, automatic object recognition is done via correlating features extracted from motion sensors worn by the user and attached to objects. This contributes to both monitoring during daily activities (detect which objects are handled by which user) and physical exercising (detects which exercising objects are chosen – e.g. dumbbells – and when the exercise is begun).

The methods are further detailed in the following sections.

2.1. Activity level - the IMA value

It has been shown that health condition and quality of life are directly influenced by the amount and intensity of daily physical activity [1]. This is particularly relevant to persons suffering from COPD. They enter a vicious circle, in which being active causes discomfort, making them progressively more sedentary, and deteriorating their health. Monitoring the daily activity can stimulate people to perform exercises and to be more active in general by providing feedback and assistance to better manage the physical condition.

The activity level of a person is best assessed in terms of energy expenditure. This is therefore a relevant feature for the IS-ACTIVE use cases. The IMA algorithm [3] is shown to produce a good estimate of the activity level / energy expenditure. The main steps of the algorithm are the following:

1. The accelerometer signal is first high-pass filtered to remove the gravity component of the accelerometer signal.
2. The resulting acceleration signal is then integrated according to
3. Equation 1 to obtain the IMA value. The integral produces a value that correlates with the signal energy over the three axes, which is a good measure for the intensity of the measured motion.
4. The integration time T is set to one minute. To calculate the IMA value for one hour or one day, the IMA minute values are averaged over the period involved.

$$IMA = \int_{t=t_0}^{t_0+T} |a_x|\, dt + \int_{t=t_0}^{t_0+T} |a_y|\, dt + \int_{t=t_0}^{t_0+T} |a_z|\, dt$$

Equation 1 – the IMA formula

The IMA algorithm if fully implemented and operational on the ProMove nodes, with options for configuring the time intervals to different granularity levels.

2.2. Exercising - detection of the direction of vertical motion

Assisted exercising is one of the objectives of the IS-ACTIVE project. Detailed descriptions of exercises for COPD patients are provided in D2.1 – Requirements. We aim to stimulate patients to exercise more by means of enhanced games, where sensors are embedded in objects and the game reacts to the motion of the user and objects. A concrete example is dumbbell lifting, where a ProMove sensor node in embedded in the dumbbell and recognizes automatically the up-down movements.

This game scenario poses the challenge of detection of the direction of vertical motion – up or down – regardless of the position or orientation of the dumbbell. This is a significant challenge compared for example to the now widely used Wii system, where the Wiimote is always held and handled in a certain position (by design).

We researched and implemented therefore an algorithm for up-down feature detection using the ProMove on-board accelerometer sensor. The objectives of the algorithms are:

- To obtain the vertical component of dynamic acceleration, removing the influence of gravity, and subsequently detect the upward or downward motion.
- To be orientation-independent and work even if the sensor orientation changes during movement (for example if the user tilts the dumbbell during the exercise).

To obtain the vertical component of dynamic acceleration, removing the influence of gravity, we project the acceleration vector on the estimated gravity vector. This is achieved by calculating the dot product between the gravity vector and the sensor acceleration [2], as shown in Equation 2.

$$\vec{p} = \frac{\vec{a} \cdot \vec{g}}{\vec{g} \cdot \vec{g}} \vec{g}$$

Equation 2 – Vertical projection of acceleration

The resulting projection ***p*** is parallel to gravity ***g***. The dot product yields a scalar result and thus the fractional part before ***g*** in Equation 2 represents a scale value, indicating the magnitude of the projection relative to ***g*** itself. Since we are mainly interested in the sign of the projection relative to ***g***, we can simply use the numerator of the scale value, yielding the decision function from Equation 3.

$$D = \begin{Bmatrix} 1 & up & if\, \vec{a} \cdot \vec{g} > 0 \\ 0 & none & if\, \vec{a} \cdot \vec{g} = 0 \\ -1 & down & if\, \vec{a} \cdot \vec{g} < 0 \end{Bmatrix}$$

Equation 3 – Decision function

Figure 1 shows idealized examples of the projected vertical acceleration and integrated vertical velocity resulting from this algorithm. Both upwards and downwards movement are shown. As the figure shows, the acceleration is first in the direction of the movement during acceleration and subsequently against it. Compared to the idealized example, the real-life sensor signals are much more noisy, noise that results directly in errors in the correct “up” or “down” classification. That is why we integrate the acceleration into a vertical velocity estimate, which also smoothens much of the noise. As shown in the bottom plot of Figure 1, this results in a single peak either going up or down.

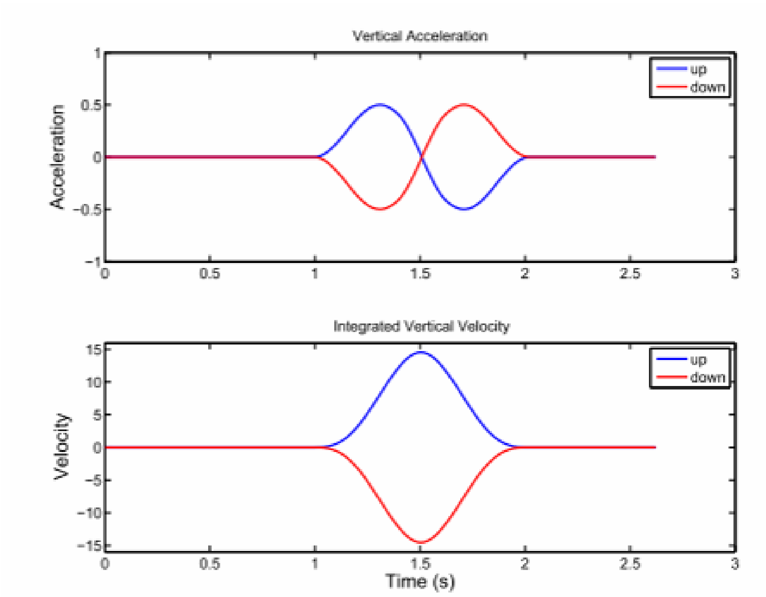

Figure 1 - Ideal example of projected vertical acceleration and integrated vertical velocity of upwards and downwards motion

The implementation is shown schematically in Figure 2. The movement of the user is sensed by the accelerometer. Using the sensor output, the gravity vector is estimated. The estimated gravity vector is used to extract the current sensor acceleration from the acceleration measurement and to project that on to the gravity vector as explained. The resulting projection is saturated to remove excessive spikes, which are caused by impact, e.g. when the sensor is tapped. Also, the projection is clamped to zero if it is below a certain threshold to prevent noise from affecting the next step. The next step involves integrating the vertical acceleration into a vertical velocity estimate. The aforementioned threshold makes sure the integral is not changed by noise when the sensor is still. The final step detects the spikes in the velocity to distinguish upwards and downwards motion. The integral is reset to zero when a peak has passed.

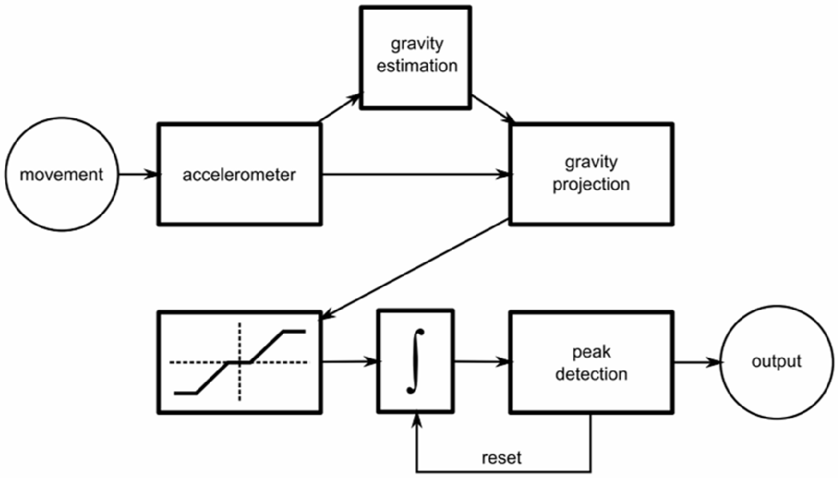

Figure 2 - Overview of implementation of up-down movement detection

2.3. Extracting features for automatic recognition of object handling

User interaction with smart objects represents an important component of the ambient assisted living vision. As debatable as the definition of a “smart object” may be, it is logical to assume that a smart object should, at the very least, detect when the user is interacting with it. A more advanced piece of functionality would be to additionally recognize how the user is interacting with it and further on, by combining such information from various objects, to infer the user’s activities and provide specific support.

The IS-ACTIVE system accurately detects what objects the user is interacting with by equipping the objects of interest and the user’s arm with wireless sensor nodes that measure their motion using inertial and magnetic sensors [4]. In order to detect object use, relevant features of the collected motion data from the objects and the user’s arm are correlated. The feature extraction, communication and correlation is done on-line and cooperatively by the sensor nodes, which guarantees a fast response time to the user.

The feature values are extracted from the raw signal at regular non-overlapping intervals called windows. A new feature is computed at the end of such an interval, which means that the length of the windows, i.e. the window size W, determines the rate at which features are generated with a given sample frequency fs, i.e. the feature frequency (ff = fs/W).

The extracted features need to meet certain requirements: (1) the extracted features must adequately retain the overall motion characteristics, (2) the processing requirements need to be as low as possible, as dictated by the resource limits of sensor node hardware, (3) the extracted features must be small and produced at a low rate to keep bandwidth and processing requirements low, and (4) the features must be computed such that the absolute orientation of the sensor has little or no influence on the produced feature result.

To meet these requirements, we define a feature vector composed of two features that describe the intensity of lateral and rotational movements and are orientation independent.

2.3.1. Compass rotation angle

We infer the intensity of rotation during a given time interval by calculating the angle between vectors measured at the beginning and the end of a feature window through the dot product. The compass rotation angle f_{cra} feature is calculated from the compass measurements $\vec{m} = \langle m_x(t), m_y(t), m_z(t) \rangle$ in the window interval $t = 1...N$ as follows:

$$f_{cra} = \frac{\vec{m}(1) \cdot \vec{m}(N)}{|\vec{m}(1)||\vec{m}(N)|}$$

Equation 4 - The compass rotation angle

The feature value is normalized to yield a cosine angle value in the interval [−1; 1]. When there is no rotation, $f_{cra} = 1$.

2.3.2. Mean acceleration magnitude

To make the accelerometer data insensitive to the current orientation of the sensor, two important steps are taken within a window interval $t = 1...N$:

1. The raw accelerometer signal $\vec{a_r}(t) = \langle a_{r,x}(t), a_{r,y}(t), a_{r,z}(t) \rangle$ is stripped from its offset by subtracting the mean value in the window interval. This coarsely compensates for the gravity component, which is orientation-dependent and usually changes slower than the actual acceleration the sensor is subjected to. Additionally, this step compensates for the effect of offset miscalibration, avoiding the need to calibrate the individual accelerometers.
2. The sum of the absolute vector components is calculated from the three axis components. This has a similar response to the more computation intensive acceleration magnitude. This discards the vector's direction and retains only its length, resulting in one value per sample that describes the desired intensity.

Summarizing, this mean acceleration magnitude f_{mam} feature is calculated from the raw acceleration samples in the window interval $t = 1...N$ as follows:

$$a_x(t) = a_{r,x}(t) - \overline{a_{r,x}}$$

$$a_y(t) = a_{r,y}(t) - \overline{a_{r,y}}$$

$$a_z(t) = a_{r,z}(t) - \overline{a_{r,z}}$$

$$f_{mam} = \frac{1}{N} \sum_{t=1}^{N} |a_x(t)| + |a_y(t)| + |a_z(t)|$$

Equation 5 - The mean acceleration magnitude

For a pair of sensor nodes under consideration, the motion features are correlated in the time domain using the Pearson product-moment correlation coefficient. This gives an indication of whether the two nodes are moving the same – i.e. the node on the user's arm and the smart object are moving together. A node performs the correlation calculation using its local measurements and those communicated wirelessly from the peer node. The correlation is calculated over the last H features produced at both nodes. The result lies in the range [−1; 1], for which a value of 1 means that the signals are fully correlated and values less or equal to 0 mean that the signals are not correlated.

3. Conclusions

The first prototypes of the feature extraction methods are available to the IS-ACTIVE consortium, being implemented on the ProMove platform. The methods cover both usage scenarios required by the project: long-term daily activity monitoring and short-term physical exercising.

4. References

[1] Bouten, C.: Assessment of daily physical activity by registration of body movement. Eindhoven, The Netherlands: Eindhoven Univ. Technol (1995), pp. 145

[2] D. Mizell, "Using gravity to estimate accelerometer orientation," in Wearable Computers, 2003. Proceedings. Seventh IEEE International Symposium on, 2003, pp. 252–253.

[3] Bosch, S. and Marin-Perianu, M. and Marin-Perianu, R.S. and Havinga, P.J.M. and Hermens, H.J. (2009) Keep on Moving! Activity Monitoring and Stimulation Using Wireless Sensor Networks. In: Smart Sensing and Context: 4th European conference, EuroSSC, 16 Sept - 18 Sept 2009, Guildford, United Kingdom. pp. 11-23. Lecture Notes In Computer Science 5741. Springer Verlag.

[4] Bosch, S. and Marin-Perianu, R.S. and Havinga, P.J.M. and Marin-Perianu, M. and Horst, A.P. and Vasilescu, A. (2010) Automatic Recognition of Object Use Based on Wireless Motion Sensors. In: International Symposium on Wearable Computers 2010, 10-13 Okt 2010, Seoul, South Korea. pp. 143-150. IEEE Computer Society.