Title	Range-based algorithm for posture classification and fall-down detection in smart homecare system
Author(s)	Wongpatikaseree, Konlakorn; Lim, Azman Osman; Tan, Yasuo; Kanai, Hideaki
Citation	2012 IEEE 1st Global Conference on Consumer Electronics (GCCE): 243-247
Issue Date	2012-10
Type	Conference Paper
Text version	author
URL	http://hdl.handle.net/10119/10926
Rights	This is the author's version of the work. Copyright (C) 2012 IEEE. 2012 IEEE 1st Global Conference on Consumer Electronics (GCCE), 2012, 243-247. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Description	

Range-based Algorithm for Posture Classification and Fall-down Detection in Smart Homecare System

Konlakorn Wongpatikaseree, Azman Osman Lim, Yasuo Tan
School of Information Science
Japan Advanced Institute of Science and Technology
Ishikawa, Japan
{w.konlakorn, aolim, ytan}@jaist.ac.jp

Hideaki Kanai
Center of Knowledge Science
Japan Advanced Institute of Science and Technology
Ishikawa, Japan
hideaki@jaist.ac.jp

Abstract— **Human posture classification is one of the most challenging issues in smart homecare system. To achieve high classification accuracy, we propose a new algorithm, called range-based algorithm. In this paper, a range means the distance between body parts. The ranges between body parts are investigated to classify the human posture and to detect a possible fall-down accident. Furthermore, we also proposed an adaptive posture window scheme to recognize the human posture in real-time even though human change the posture in different speed. The results reveal that our proposed can classify the human posture and detect fall-down with high accuracy and reliability.**

Keywords-component; Human posture classification, fall-down detection, range-based algorithm, smart homecare system.

I. INTRODUCTION

A growing elderly population and strong demand for efficient in-home healthcare have fueled a growth in smart homecare system, which may assist residents by providing in-home nursing assistance and emergency communication. In smart homecare system, we propose a human monitoring service platform, which can be also used for the applications of human behavior monitoring and human positioning. The proposed platform architecture is composed of three parts: sensing, algorithm, and database. In this paper, we are intended to focus on the algorithm part. Therefore, we propose a novel range-based algorithm for posture classification and fall-down detection in smart homecare system. In particular, sensors that are placed at body parts will allow vast amounts of the precise 3D sensor's location data to be collected automatically. Our range-based algorithm uses these collected data for classifying the mined data of human posture. Whereas algorithm, which uses only user's height information, leads to an inefficient of posture classification, e.g., when a user sit on the table, the user's height is the same as user standing. However, the range-based algorithm uses the range between the body parts to clearly classify the human posture whether the user is sitting on the table or s/he is standing. In summary, the range-based algorithm is vital and essential for human posture classification.

In this research, our primary goals are to classify the human posture with high accuracy and reliability to yield high-confidence data suitable for in-home nursing assistance and to handle the emergency situation in real-time. Our main contributions are the following. First, we are proposing the range-based algorithm to classify the human postures such as sitting, standing, or lying down using three ultrasonic sensors. Second, we are introducing the finite state machine to detect the unexpected fall-down accident. Third, we are addressing the adaptive posture window scheme to capture the human posture in different speed of user. In this research, we also include a height-based algorithm for comparison purpose. Our experimental results reveal that the range-based algorithm outperforms the height-based algorithm in terms of posture classification and fall-down detection with high accuracy and reliability.

This paper is organized as follows: in section II, we briefly describe background and related works. Section III explains our proposed algorithm. Experiments and results are addressed in section IV. After that, we discuss some implementation issues in section V. Finally, we conclude the overall of this research and also future direction of this research in the last section.

II. BACKGROUND AND RELATED WORKS

A. Background

Before implementing the posture/activity classification system, we have to clear understanding between two words: posture and activity. Posture is the position of the body parts such as standing, sitting, or lying down. Whereas activity is the something that human do or cause to happen, such as walking, watching TV, or sleeping. In consequence, sometimes we can express that posture is a subset of activity and some researches combined postures and activities classification into one system. However, this paper will focus only on posture classification.

Until now, activity monitoring has been proposed for recognition the human activity in home from several years [1, 2]. It is necessary to know what the home user does in each day because the activity information can be used for several purposes such as service recommendation, healthcare system and etc. Furthermore, the other useful information is generated from the activity information such as activity of daily life (ADL) [3], human behavior [4], or healthcare information [5]. Nevertheless, classifying the human activities is not an easy task because we encountered many problems such as environment, accuracy, privacy and so on. It still needs a good technique to improve the system efficiency.

B. Related Works

In this section, we are going to describe several techniques that are used for activity monitoring. There are three areas for classification.

1) Computer Vision

Computer vision is one area that uses a camera to extract the activities of the user in home. Image processing technique is, therefore, used to analyze the human activities on the picture. Jansen, et al. [6] proposed the technique that used 3D camera to recognize the human poses in the home. They applied the depth information to track where the human was in image, then classified the human poses by using the pose recognition algorithm. Furthermore, the camera was used not only to detect the human action, but also to find the appearance and travel time of human. Zou, et al. [7] presented the distributed camera network for classification the human activities. They separated the human activities into four different classes: normal, break-in, stay, and sudden appearance/disappearance. This information is necessary to monitor the old people, who live alone in the home. However, using the camera for recording the activities all the time may lead to privacy issue because the user may be threaten his/her personal life.

2) Home Sensor Network (HSN)

The advantage of this technique is that it does not need to attach any sensors on human body, but several sensors are required in the HSN technique to built in the home facility such as toilet, TV, and bed. The system perceives the human activities by monitoring, what the home appliance is in use and how long user spends time on appliances [8].

In fact, only the home sensor data can classify the human action in the home, but it tends to be low accuracy and confusing in case of many home appliances are being used at the same time. Moreover, this technique requires a large number of sensors for recognition the human activities.

3) Body Sensor Network (BSN)

The concept of the BSN technique is to use the sensor attached on the body part of user in order to collect the various kinds of information such as vital sign, energy, or correlation of acceleration data. Accelerometer sensor has been used most frequently in the BSN area. For example, five small biaxial accelerometer sensors were used in [9] to classify 20 activities. The results showed the recognizing performance in 84%. Lee, et al. [10] also proposed the three-axis accelerometer sensor to recognize the human activities and recorded in the personal life log. However, the problem of the BSN technique is that it is difficult to analyze raw data. In addition, it is also difficult to discern body posture when the person does not move, which leads to undetectable signals for identifying the posture.

The difficult task of the BSN technique is not only in the feature extraction, but also in the Artificial Intelligent algorithm. It needs to use the efficient algorithm such as decision tree [1], HMM [2], fuzzy logic [5] and so on to classify the activities.

III. PROPOSED ALGORITHM

A. Binary Decision Tree

A binary decision tree is applied to distinguish the human postures. It consists of five nodes: three leaf nodes represent the human posture and two parent nodes represent the binary decision, which use a threshold technique to distinguish the human posture. The *threshold value* is defined by the proportion range between the body parts. For example, Fig. 1 presents the binary simple decision tree. T_1 is defined by range value between hip and knee, and range value between knee and shoulder is determined as a threshold value in T_2. If the range value between hip and knee is higher than T_1, the posture will be “Stand”, or if the range value between knee and shoulder is higher than T_2, the posture will be “Sit”.

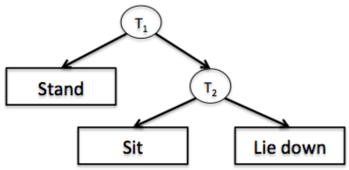

Fig. 1 Binary Decision Tree for posture classification.

B. Finite State Machine (FSM)

We use a FSM to handle the current user posture state. We define two states. First, “*posture state*” is only one fundamental posture. Second, “*changing posture state*” is the transition from one posture to another posture, as presented in Fig. 2

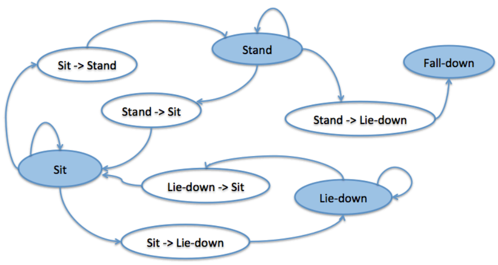

Fig. 2 Finite state machine.

Fig. 2 illustrates the FSM of this system, it consists of nine states: five changing posture states and four posture states. The posture states can be divided into two types: human postures (sit, stand, lie-down) and unexpected accident (fall-down). In the changing posture states, we do not have the “Lie-down→Stand” state because in the real human behavior, when human changes the posture from lying down to standing, the posture of human will be sitting before standing. In the same way, “Stand→Lie-down” state can be defined as fall-down because it does not perform the sitting posture before lying down.

C. Adaptive Posture Window Scheme

Because the computation time of our posture classification system is real-time, the outcomes of the classification are changing rapidly. The adaptive posture window is adopted to identify the current posture. The adaptive posture window starts with a *subsequence posture* in the time series. The subsequence posture in the adaptive posture window is the "*posture state*". Therefore, time computation (*TC*) is defined as below:

$$TC = Window_{size} \times Time\ sampling \quad (1)$$

where $Window_{size}$ is the size of adaptive posture window, which is depending on the environment such as the speed of changing posture and so on. *Time sampling* is an interval time for each "*posture state*". In this paper, size of adaptive posture window is fitted depending on posture's speed because when human change the posture, speed for changing the postures is not a same. Types of human also affect to the speed for changing the postures such as old people, children, or average person. For this reason, adaptive posture window scheme is implemented in order to improve the posture classification's accuracy and reliability.

D. Posture Pattern Recognition (PPR)

From the adaptive posture window scheme, there are three fundamental patterns can be obtained: *majority pattern*, *equal pattern*, and *minority pattern*. We use two methods; a *ratio method* and a *transition weight method* to recognize a resultant state, which is defined as the current "*posture state*" in FSM. The ratio method is used to justify the posture patterns. The posture patterns are recognized by counting the number of "*posture states*" in the adaptive posture window. The majority pattern is recognized if the number of "*posture state*" more than half of the size of adaptive posture window, the resultant state will be the corresponded human posture as shown in No.1, 2 in Table 1. While, if there are at least two postures having a same number, we will classify that is an equal pattern and the rest condition will be a minority pattern. In majority pattern, we do not need to use the transition weight (TW) method because this pattern can be inferred to exactly posture by itself. Nevertheless, equal and minority patterns need TW method to decide the resultant state. We take the transition weight (TW) value, in the transition between two postures in the adaptive posture window as +1, +2, +3, and so on for the postures between first and second, second and third, third and four, and so on, respectively. The TW value will be added when posture changes. Threshold technique is also combined with the TW method. Threshold of TW method will be changed depending on the size of adaptive posture window. Fig. 3 describes the flowchart for decision of resultant state, which size of posture window is 4.

Table 1. The example of posture patterns with size of adaptive posture window of 4.

No	Posture window	Pattern	Transition weight	Resultant state
1	Stand-Stand-Stand-Stand	Majority	-	Stand
2	Stand-Sit-Stand-Stand	Majority	-	Stand
3	Stand-Stand-Sit-Sit	Equal	2	Stand→Sit
4	Stand-Sit-Stand-Sit	Equal	6	Sit
5	Sit-Lie-down-Stand-Stand	Minority	3	Sit→Stand
6	Stand-Stand-Sit-Lie-down	Minority	5	Stand

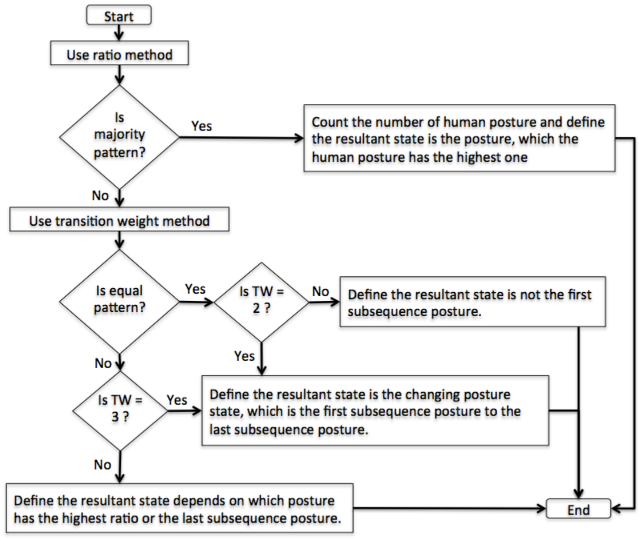

Fig. 3 Flowchart for PPR when size of adaptive posture window is 4

E. Range-based Algorithm

We observe the postures from the real human action in daily life, and setup the hypothesis that each posture has different physical pattern. This means the relation between body parts conform to the human posture. Thus, concept of this algorithm is determined the relation of body parts, and extracting the postures from the range between body parts. We attached three sensors on shoulder, hip, and knee to perform the range-based algorithm.

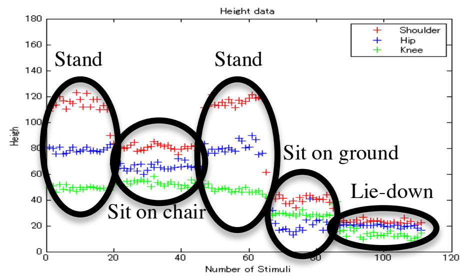

Fig. 4 Range between body parts in each posture.

Fig. 4 shows that the ranges between body parts are extracted from the height data, and it will increase/decease in y-axis depending on the postures. For example, standing and sitting have different range between shoulder and knee in y-axis, whereas lie-down has a little difference in range value between body parts in y-axis. Thus, we can distinguish the postures by measuring the range between the sensors.

F. Height-based Algorithm

The idea of the height-based algorithm uses one sensor attached on the user's shoulder to classify the human postures. The decision tree of the height-based algorithm is similar to the range-based algorithm, but it is different as to the threshold value. The threshold is defined by the shoulder's height of the user. Meanwhile, the adaptive posture window, the posture pattern recognition, and the FSM are same as above.

IV. EXPERIMENTS AND RESULTS

A. Experimental Setup

The proposed range-based algorithm and the height-based algorithm are designed and installed in AwareRium [11], a room for experimental environment to investigate various support systems. The installed sensors that are ultrasonic sensors can detect and monitor the position information of user in AwareRium. Although Bao and Intille have proposed an activity classification using an accelerometer sensor [9], both range-based and height-based algorithms are still applicable because they do not depend on any type of sensor. However, our proposed algorithm differs from the existing research. In this paper, the height-based algorithm is used to compare with the range-based algorithm because they use the same resources. The comparing procedure shows in Fig. 5. Accuracy in our results is defined as the ratio of the corrected number of posture states and changing posture states over the total number of resultant states in this experiment.

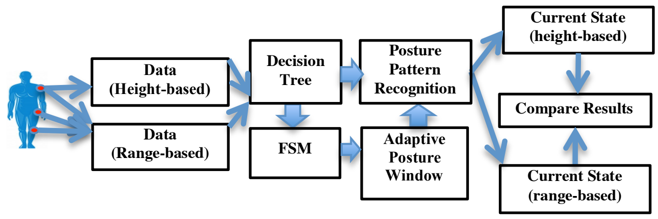

Fig. 5 Procedure of comparison between range-based algorithm and height-based algorithm.

B. Experimental Results

In first experiment, we develop the adaptive posture window scheme and adjust the size of adaptive posture window depending on the speed of changing posture. Size of adaptive posture window is changing from 3 to 6 posture states. In this experiment, we define time sampling to 0.2 seconds. We evaluate the change of size of posture window for an average person, shown in Fig. 6.

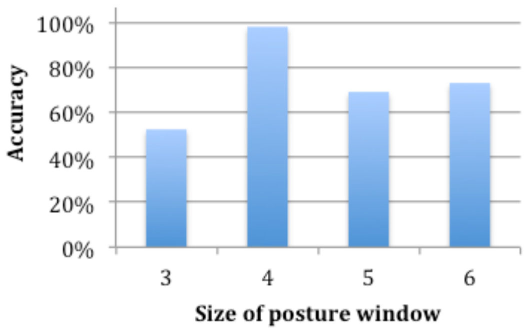

Fig. 6 The difference between size of posture windows and accuracy.

After several experiments, the optimum size of adaptive posture window is four. This size of adaptive posture window can achieve the highest accuracy in our environment scenario, which is performed by one male and one female subjects (age ranged between 25-26, height between 164cm and 174cm). However, the other sizes can be used in further circumstances. For example, size of adaptive posture window of 6 is suitable for old people. Because time computation of this size is around 1.2 seconds, it collects six subsequence postures inside. It quit easy to find the current posture if the people change the posture very slow as old people. On the other hands, if the people change the posture very fast as children, the inside window will hold various kind of posture states. It is very hard to decide the current posture. Thus, small number of windows is suitable for whom moving fast.

In second experiment, we setup the experiment to verify the problem when a user is merely moving. This means that user performs only one posture in one period of time. The test subject performs each posture in five minutes and five times per test set. We also design the object's height that the test subject sits or lie-down. The results are shown in Table 2.

Table 2. Accuracy of height-based algorithm and range-based algorithm in static posture experiment.

Posture	Height-based	Range-based
Stand	100 %	100 %
Sit *	100 %	100 %
Sit **	0 %	100 %
Lie-down ***	100 %	100 %
Lie-down ****	0 %	100 %

* Object that the user sitting is lower than hip's height.
** Object that the user sitting is higher than hip's height.
*** Object's height that the user lying is lower than 15 cm.
**** Object's height that the user lying is higher than 15 cm.

In third experiment, we focus on the consequence postures experiment by looking at the "*changing posture state*" in FSM. We setup the consequence postures scenario as "Lie-down→Stand→Sit→Stand→Lie-down→Sit→Stand", and perform 10 times. These consequence postures can cover all of the states in FSM. Table 3 shows the accuracy in the consequence postures experiment.

Table 3. Accuracy of the consequence postures experiment.

Consequence postures	Height-based	Range-based
Lie-down→Stand→Sit→Stand→ Lie-down→Sit→Stand	86 %	98 %

According to the results of the consequence posture experiment, the main error of both algorithms occurs in state "Lie-down→Sit". Since sitting posture in our experiment means sit on the chair. If we consider real human behavior, order of the posture should be "Lie-down → Sit(ground) → Stand → Sit(chair)". The height-based algorithm often classifies the state only in "Lie-down→Sit(ground)" which is incorrect, whereas the range-based algorithm uses three data for classification, which makes classification more precisely. It has a bit error when moves very fast. In addition, the results of the second and third experiments show that, the posture in the height-based algorithm is not flexible compared to the range-based algorithm. For example, standing posture in the height-based algorithm has to stand up straight, user cannot bend down, while user can do both in the range-based algorithm.

In the last experiment, a fall-down detection, it is extremely important for monitoring the unexpected situation, especially old people who live alone in the home. Fall-down is defined as a backward fall-down. In this experiment, fall-down is examined by detecting the "*changing posture state*" of "Stand→Lie-down". Table 4 shows the accuracy of the correct classification for the fall-down detection in five times.

Table 4. Accuracy of the fall-down detection experiment.

Algorithm	Accuracy of fall-down detection
Height-based	80 %
Range-based	100 %

V. DISCUSSIONS

Although the range-based algorithm uses the simple concept to classify the human postures in home, the results are efficient. The postures in the range-based algorithm are more flexible than the height-based. For example, when a user is bending over the table, the shoulder's height reduces over the threshold. The range-based still recognize the posture as "Stand" by using range between hip and knee, but the height-based will classify as "Sit". Thus, we can see that the important thing to classify the postures with this algorithm is a relation of body parts. Moreover, this algorithm is not fix to the type of sensor.

From the results in our experiments, the range-based algorithm demonstrates the simple technique for classification, but achieves the high accuracy when compare with the existing techniques. Most of existing researches have a good accuracy in dynamic activity, but results in static activity or posture classification, are quite low. One thing that makes the results between our proposed technique and existing technique difference is type of data. In our experiment, we used ultrasonic sensors that can produce the height data. We can use this data for extracting the range between body parts, and ultrasonic sensors achieve stable and accurate data. On the contrary, the existing techniques were used the accelerometer sensor that the output of sensor is a raw signal. It needs some feature extraction to gain the necessary variables. [10] is a good example research to explain the problem in accelerometer sensor. They obtained high performance in dynamic activity classification, 90.65%, while performance in static activity classification drop to 83%. In their work, the error can be found easily in static activity because signals are quite stable. It is different in terms of magnitude is each axis. Feature extraction cannot be used very well if signal input is a stable signal. On the other hand, signals in dynamic activities have oscillations. It is easy to extract the value from the oscillation signals. As a conclusion accelerometer is not so accurate for posture classification.

VI. CONCLUSION

In this paper, we have proposed the range-based algorithm to classify human posture and to detect fall-down accident for smart homecare system. The range-based algorithm is to measure the distance between body parts, and then extract the range relation between body parts to classify the human posture. Moreover, the adaptive posture window scheme has also developed to select the appropriate size of adaptive posture window regardless of the speed of changing posture. From our experiment, the test subject is an average person that performs all postures normally, general speed. The system adopts the size of posture window to 4 for posture classification. The advantages of our proposed algorithm are high accuracy, high reliability, and real-time operation. Nevertheless, this proposed algorithm can distinguish only the human postures. It still needs more techniques for classification in various cases.

Our future work, the range-based algorithm has been required additional techniques to classify more specific activity such as watching TV, playing computer, or cooking. Thus, we plan to use other techniques, the HSN technique, to classify more specific activity. We also plan to use this information for analyzing the human behavior in the healthcare system.

REFERENCES

[1] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola and I. Korhonen, "Activity Classification Using Realistic Data From Wearable Sensors," IEEE Transactions on Information Technology in Biomedicine, Vol. 10, No. 1, pp. 119-128, January 2006.

[2] J. He, H. Li and J. Tan, " Real-time Daily Activity Classification with Wireless Sensor Networks using Hidden Markov Model," In 29th IEEE EMBS 2007, pp. 3192-3195, Lyon, France, August 2007.

[3] S. Bang, M. Kim, S. Song and S. Park, "Toward Real Time Detection of the Basic Living Activity in Home Using a Wearable Sensor and Smart Home Sensors," In 30th IEEE EMBS 2008, pp. 5200-5203, Vancouver, Canada, August 2008.

[4] M. Nii, K. Nakai and Y. Takahashi, "Extraction of Human Behavior from Human Activity Monitoring Data using MEMS Sensors," In IEEE Workshop on Robotic Intelligence in Informational Structured Space, pp. 7-11, Paris, France, April 2011.

[5] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu and P. Havinga, "Activity Recognition Using Inertial Sensing for Healthcare, Wellbeing and Sports Applications: A Survey," In 23th ARCS 2010, pp. 167-176, Hannover, Germany, Febuary 2010.

[6] B. Jansen, F. Temmermans and R. Deklerck, "3D human pose recognition for home monitoring of elderly," In 29th EMBS 2007, pp. 4049-4051, Lyon, France, August 2007.

[7] X. Zou and B. Bhanu, "Anomalous Activity Classification in the Distributed Camera Network," In ICIP 2008, pp. 781-784, California, USA, Ocotober 2008.

[8] S. Zhang, S. McClean, B. Scotney and C. Nugent, "Using duration to learn activities of daily living in a smart home environment," In 4th International Conference on Pervasive Computing Technologies for Healthcare 2010, pp. 1-8, Munich, Germany, March 2010.

[9] L. Bao and S.S. Intille, "Activity recognition from user-annotated acceleration data," *In Pervasive Computing*, pp.1-17, Vienna, Austria, 2004.

[10] M. Lee, A. M. Khan, J. Kim, Y. Cho and T. Kim, "A Single Tri-axial Accelerometer-based Real-time Personal Life Log System Capable of Activity Classification and Exercise Information Generation," In: 32nd EMBS, pp. 1390-1393, Buenos Aires, Argentina, September 2010.

[11] H. Kanai, T. Nakada, G. Turuma, and S. Kunifuji, "An aware-environment enhanced group home," *Springer Lecture Notes in Computer Science*, vol. 4413, pp.425-436, 2007.