

Activity Recognition with Multi-tape Fuzzy Finite Automata

H. Karamath Ali

Associate Professor in Computer Science, Periyar EVR College (Autonomous), Tiruchirappalli, India
Email: hkaramath@yahoo.com

Dr. D. I. George Amalarethinam

Associate Professor & Director-MCA, Jamal Mohamed College (Autonomous), Tiruchirappalli, India
Email: di_george@jmc.edu

Abstract — Recognizing the activities performed by the user in an unobtrusive manner is one of the important requisites of pervasive computing. Users perform a number of activities during their day to day life. Tracking and deciding what a user is doing at a given time involves a number of challenges. The lack of a precise pattern in doing an activity at different times is one among them. The number, order, and duration of the different steps involved in an activity vary significantly, even when the activity is done by the same user at different times. To overcome these challenges, a number of simultaneous inputs have to be handled with provisions for handling variations in number, order and duration of these inputs. This paper explains how multi-tape fuzzy finite state automata can be used to effectively recognize human activities. The method explained is found to give good results when tested using publicly available activity datasets collected in a smart home environment.

Index Terms — Pervasive Computing, Activity Recognition, Fuzzy Automata

I. INTRODUCTION

The ability to recognize human activities automatically and unobtrusively in a smart environment has a number of applications varying from essential services such as healthcare and security to luxurious services such as automatically adjusting room ambience and ergonomics[1][2][3]. A smart environment usually has a large number of different types of sensors embedded in almost every possible component of the environment. These sensors provide input about the user's interaction with the environment. Using these inputs the activity recognition system has to decide and possibly predict the activity performed by the user. This decision or prediction may then be used by some other larger system to provide appropriate services to the user. The task of activity recognition gets complicated by the

need to handle many simultaneous sensor inputs and the impreciseness with which users perform activities. A number of probabilistic and structural methods have been developed by researchers to recognize activities [4] - [19]. This paper illustrates how multi-tape fuzzy finite automata can be used for recognizing human activities. The multi-tape feature will be useful in representing multiple simultaneous inputs; and the concept of fuzziness can be used to represent and deal with the variations in the pattern of human activities. The method is found to produce results that are comparable to other frequently used methods.

The remainder of this paper is organized as follows: Section II presents an overview of the related works in activity recognition; section III defines the problem statement; section IV presents the fundamental concepts of multi-tape automata and fuzzy automata; section V explains the proposed method; section VI discusses the dataset used and the experiment conducted; and section VII presents conclusion.

II. RELATED WORK

In their attempt to develop a framework for solving the problem of activity recognition, researchers have used a number of models. Among them, the Hidden Markov Model (HMM) is the most commonly used. Sanchez, D. et al. [1] trained a discrete HMM to map contextual information to a user activity. The model was trained and evaluated using data captured from 200 hours of detailed observation and documentation of hospital workers. Tim van Kasteren et al. [2] recorded a dataset consisting of 28 days of sensor data in a smart home environment and through a number of experiments showed how the HMMs perform in recognizing activities. An approach for multi-person activity recognition in an office environment using audio as well as video features, employing a multilevel HMM framework is used by Christian Wojek et al. [3]. Human activities embedded in video sequences acquired

in an archeological site are automatically recognized using Discrete HMMs by Marco Leo, et al. [4]. Weiyao Huang, et al. [5] used Discrete HMM for human posture training, modeling and activity matching to recognize human motion. Other HMM variations such as Coupled HMM, Fuzzy HMM, Hierarchical HMM and Reconfigurable HMM are used by other researchers [6] [7] [8] [9].

Roc óD áz de León et al.[10] used a Bayesian network for recognition of continuous activities, by considering the direction change between frames to track the motion of several limbs and to recognize different activities. An approach for human activity recognition that can recognize activities performed at different velocities by different people and can work with missing data, based on the Fourier transform and Bayesian networks was presented by Roc óD áz de León [11]. A system that uses naïve Bayesian classifier for recognizing activities in the home setting using a set of small and simple state-change sensors is introduced by E.M. Tapia, et al. [12]. In the framework suggested by Sangho Park and J.K. Aggarwal [13], human action is represented in terms of verbal description according to *subject + verb + object* syntax, and human interaction is represented in terms of *cause + effect* semantics between the human actions. Then, a dynamic Bayesian network is constructed to estimate temporal evolution of the human actions for recognizing the dynamic gestures of the body parts. Tim van Kasteren and Ben Kr öse [14] used a Dynamic Bayesian Network, to model the temporal aspects of the activities of daily living (ADL) of elders. They observed that adding more sensors does not necessarily increase accuracy. A multi-level dynamic Bayesian network was used to perform complex event recognition by Justin Muncaster and Yunqian Ma [15]. The levels of the network are used to enforce state hierarchy while the lowest level models the duration of simplest event. The probability that a scenario occurs is computed from the mobile object properties obtained from a sequence of image frames using several layers of naïve Bayesian classifiers in the work presented by Somboon Hongeng, et al. [16].

Friedrich Steimann and Klaus-Peter Adlassnig presented a framework for an intelligent bedside monitor that derived an abstraction of the current status of a patient by fuzzy state transitions on pre-processed input supplied continuously by clinical instrumentation [17]. This is one of the earlier works that use fuzzy automata for activity recognition. A Fuzzy Rule based Classifier and two Fuzzy Finite State Machines (FFSM) are used by A. Alvarez-Alvarez, et al.[18] to recognize activities like working in the desk room, crossing the corridor, having a meeting, etc. The classifier is used to get an approximate position at the level of discrete zones such as office, corridor and meeting room. One of the FFSM is used for human body posture recognition and the other FFSM combines the localization and posture recognition. The use of fuzzy finite state

systems is demonstrated in [19] for human gait modeling. Gonzalo Bailador and Graci án Trivi ò [20] propose a syntactic pattern recognition approach based on fuzzy automata, which can cope with the variability of patterns by defining imprecise models. The approach is called temporal fuzzy automata as it allows the inclusion of time restrictions to model the duration of the different states. This approach is used for recognizing hand gestures.

Though the above works use fuzzy automaton to deal with variability and impreciseness of human activities, the states of the fuzzy automata are manually defined by an expert. This will be tedious, as the number of states becomes very large. So in this paper we propose to use an algorithm suggested by Henning Fernau [21] to automatically construct a DFA to recognize human activities using a set of sensor inputs, represented as strings. Then the concept of fuzziness is introduced by using appropriate functions to deal with variability and impreciseness of human activities. The proposed method is tested with a publicly available data set, and the results are comparable to other commonly used methods.

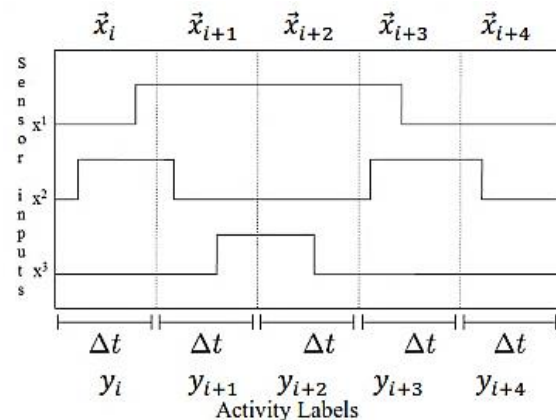


Figure 1. Sensor readings and activity labels

III. THE PROBLEM

The objective is to recognize activities from sensor readings in a smart environment. For this, the time series data of sensor readings is divided into time slices of constant length. Each time slice is labeled with the activity performed during that time slice [2]. A vector $\vec{x}_t = (x_t^1, x_t^2, \dots, x_t^N)^T$ is used to represent the sensor readings at time slice t , where x_t^i represents the input from the sensor x^i during the time slice and 'N' is the number of sensors. The activity performed during time slice 't' is represented by y_t . So, the task of the activity recognition system is to find an association between a sequence of observation vectors $\mathbf{x} = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n\}$ and a sequence of activity labels $\mathbf{y} = \{y_1, y_2, \dots, y_n\}$. Fig.1 illustrates the above setup, with Δt representing a time slice.

IV. MULTI-TAPE FINITE AUTOMATA

A two-way finite-state automaton with $n \geq 1$ tapes scans n read-only input tapes, each with an independent head [22]. At every step, the transition function determines the possible next states and head movements, based on the current state and the symbols currently under each head. Two special symbols $\triangleright, \triangleleft$ respectively mark the left and right ends of each input tape; $\Sigma_{\triangleright, \triangleleft}$ denotes the extended alphabet $\Sigma \cup \{\triangleright, \triangleleft\}$.

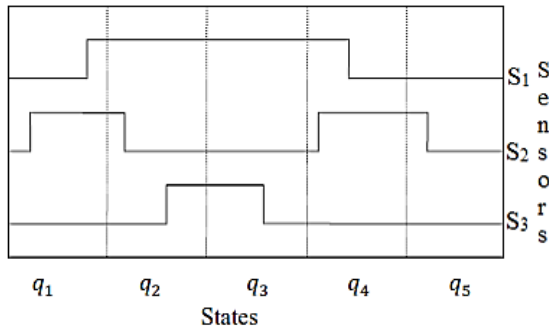


Figure 2. Time Slices and States

Definition : A two-way nondeterministic finite-state automaton with n tapes is a tuple $(\Sigma, Q, \delta, q_0, F)$, where :

- Σ -- is the input alphabet, such that $\triangleright, \triangleleft \notin \Sigma$;
- Q -- is the finite set of states;
- $\delta : Q \times \Sigma_{\triangleright, \triangleleft}^n \rightarrow \wp(Q \times \{-1, 1, 0\}^n)$ is the transition function that maps current state and input to a set of next states and head movement directions, with the restriction that the head does not move beyond the end markers;
- $q_0 \in Q$ is the initial state;
- F a subset of Q , is the set of accepting states.

The configuration of a two-way nondeterministic finite-state automaton A , is defined as a $(2n+1)$ -tuple, $\langle x_1, \dots, x_n, q, i_1, \dots, i_n \rangle \in (\Sigma^* \triangleleft)^n \times Q \times \mathbb{N}^n$ where $q \in Q$ is the current state, and, for $1 \leq k \leq n$, x_k is the content of the k -th tape and $0 \leq i_k \leq |x_k|$ is the position of the k -th head; when $i_k = 0$ (resp. $i_k = |x_k| + 1$) the head is on the left marker \triangleright (resp. right marker \triangleleft).

The transition relation \vdash between configurations is defined

as: $\langle x_1, \dots, x_n, q, i_1, \dots, i_n \rangle \vdash \langle x_1, \dots, x_n, q', i'_1, \dots, i'_n \rangle$ if and only if, for each $1 \leq k \leq n$, $\sigma_{i_k, k}$ is the symbol at position i_k in input word x_k , $\delta(q, \sigma_{i_1, 1}, \dots, \sigma_{i_n, n})$ includes a tuple $\langle q', d_1, \dots, d_n \rangle$, and $i'_k = i_k + d_k$ for each $1 \leq k \leq n$.

A run ρ of A on input $x = \langle x_1, \dots, x_n \rangle$ is a sequence of configurations $X_0 \dots X_m$ such that $X_0 = \langle x, q_0, 0^n \rangle$ and for all $1 \leq k \leq m$, $X_{k-1} \vdash X_k$. A run ρ of A on input x is accepting if $X_m = \langle x, q, i_1, \dots, i_n \rangle$ for some $q \in F$ and the i_k th character of the k -th tape is \triangleleft (that is, every head has reached the end of its tape). Correspondingly, A accepts an input word x if there is

an accepting run ρ of A on x . The language accepted by A is the set of n words $\mathcal{L}(A) = \{x \in (\Sigma^*)^n / A \text{ accepts } x\}$.

An n -tape automaton A is deterministic if $|\delta(q, \sigma_1, \dots, \sigma_n)| \leq 1$ for any $q, \sigma_1, \dots, \sigma_n$. Further, an n -tape automaton is said to be s -synchronized for $s \geq 0$ if every run of A , accepting or not, is such that any two heads that are not on the right-end marker \triangleleft are no more than s positions apart (as measured from the left-end marker \triangleright). And, A is said to be synchronized if it is s -synchronized for some $s \in \mathbb{N}$. A is simply synchronous if it is 0-synchronized. If the mapping δ never moves any of the heads left, then A is said to be one-way.

One-way, synchronous n -tape finite automata are closed under complement, intersection, union, concatenation, Kleene closure, projection, generalization, and reversal [22].

A. FUZZY FINITE AUTOMATA (FFA)

Besides a finite set of states (Q), a finite set of input symbols (Σ), and a start state (q_0), a Fuzzy Finite Automata has a finite set of output symbols (Z), fuzzy transition function (δ) and an output function (ω). $\delta : Q \times \Sigma \times Q \rightarrow (0, 1]$ is the fuzzy transition function which is used to map a current state into a next state upon an input symbol and attributing a value in the fuzzy interval $(0, 1]$ to the next state. $\omega : Q \times Z$ is the output function which is used to map a state to the output set. The membership value (mv) associated with each transition is called the weight of the transition [24].

One of the main differences between a FFA and a DFA (or an NFA) is that, in a FFA there can be more than one active state at a time. Each active state is associated with a membership value that indicates the level of activation of the state. To assign membership values to the next states, the mv of the current state and the weight of the transition are considered.

One of the characteristics of an FFA is that, a state may be forced to take several different membership values at the same time. This is called multi-membership. Detailed description of fuzzy finite automata can be found in [24].

V. THE METHOD

As explained in section III, the time series data of sensor readings is divided into time slices of equal length. Each time slice is labeled with the activity performed during that time slice. To model this arrangement using multi-tape finite automaton, input from each sensor is considered to be a tape of the automata. So, if there are n sensors to be considered in the environment, the finite automata will have n input tapes. The time slices may correspond to states of the automaton. The automaton will make state

transitions with each time slice, according to the mapping δ . The mapping rules as required may be defined, taking into account the interactions the user makes with the environment while performing an activity. This is illustrated in fig.2 in which a time series data obtained from three sensors s_1, s_2 and s_3 are given. The entire duration is divided into equal time slices and each slice is labeled with the state corresponding to the time slice (labels q_1 to q_5). For the sake of simplicity the sensors are assumed to generate binary input. Sensors that generate non-binary values may be accommodated in this model by determining suitable threshold value for each possible input stage. A one-way, synchronous 3-tape finite automata A , to model this can be constructed as follows:

$$A = (\Sigma, Q, \delta, q_0, F)$$

where

$$\Sigma = \{0, 1\}$$

$$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$$

$$q_0 \text{ is the initial state } F = \{q_6\}$$

and

$$\delta(q_0, 000) = (q_1, 1)$$

$$\delta(q_1, 110) = (q_2, 1)$$

$$\delta(q_2, 101) = (q_3, 1)$$

$$\delta(q_3, 100) = (q_4, 1)$$

$$\delta(q_4, 010) = (q_5, 1)$$

$$\delta(q_5, 000) = (q_6, 1)$$

Additional mapping conditions may be included to keep the automaton in the same state when there is no change in the inputs. Also, for each state a maximum duration may be included, after which the inputs must change to move the automaton to the next possible state; otherwise, the automaton may be moved to the initial state, so that it can be synchronized with the next possible input pattern.

In the works from the literature, the states of the fuzzy automata were decided manually by experts. This approach may not be feasible in an environment where a user may perform a single activity in a widely varied ways. The problem becomes more complicated if a large number of sensor inputs also have to be considered. So, in this work we suggest automatic construction of a finite automata that will scan the sensor input vectors and will identify the corresponding activity performed by the user. The *SL*-infer algorithms suggested by Henning Fernau [21] are used by us for the automatic construction of a finite automaton. Then, appropriate functions for deciding membership values are used to incorporate the required fuzziness in the constructed automaton.

VI. DATA AND EXPERIMENT

A number of activity datasets, collected using sensors embedded in the environment and/or worn by the subjects, are publicly available. We have used one such data set collected and made public by Tim van Kasteren, et al.[23]. The datasets have been collected by observing the behavior of inhabitants inside their homes using wireless sensor networks. Output of the binary sensors have been annotated with the activities performed by the subjects during predefined time intervals. A dataset so collected for 25 days of 10 activities such as preparing dinner, using toilet and sleeping, is used by us to conduct the experiment. Three different representations, namely raw, change point and last-fired, of the sensor data are used [23]. The raw sensor representation uses the sensor data directly as it was received from the sensors. The change point representation indicates when a sensor event takes place. The last-fired sensor representation indicates which sensor fired last. For each of the representations, six different time slices (600, 300, 60, 30, 10 and 1 seconds) have been used. Sensor data for each of the 10 activities and for each representation were culled from the data set.

A fuzzy automaton for recognizing a particular activity is constructed as follows. Each possible binary output vector in the dataset for the activity – a vector corresponding to a time slice – is assigned a unique character code. Thus a set of strings representing the time series data for the activity is generated and given as input to the DFA construction algorithm.

Fuzzy characteristic is incorporated in the generated DFA by slightly modifying the transition function. Initially the list of active states consists only of the initial state, and its membership value is set to 1.0. For each currently active state q , and the content of the input tapes at time t , x_t , the following is done. If $\delta(q, x_t)$ is defined by the generated automaton then the transition is carried out and the membership value of q is assigned to the next state. Otherwise, the distance between x_t and each x_p , for which $\delta(q, x_p)$ is defined, is calculated; this distance information is used to decide the membership value of the corresponding next possible active state. Naturally, a next state caused by an x_p with minimum distance from x_t has greater membership value than the one caused by the one with greater distance with x_t . In short, the membership value of a next state is inversely proportional to the distance of an allowable input with x_t . Lesser the distance, greater the membership value and vice versa. The problem of multi-membership is solved by choosing the maximum of the membership values.

Proceeding in this way, after the membership value of each active state for the last input in the given test string is calculated, the highest membership value of the active states is considered to decide if the given input is accepted by the FFA or not. If the highest membership value is greater than or equal to the threshold value

specified by the user, the input is accepted by the FFA, otherwise it is rejected.

The effectiveness of the automata hinges on the method for calculating the membership values of the next states. For this we have considered the membership value of a current state q and the distance between x_t and an x_p for which $\delta(q, x_p)$ is defined. The actual formula used is given below:

$$mv = (0.5 * mvc) + (0.5 * (1.0 - dist / ns))$$

where

mv – membership value of next state,

mvc – membership value of current state,

$dist$ – distance between x_t and x_p , and

ns – number of sensors in the environment.

The distance between x_t and x_p is calculated as the number of positions in which the two vectors differ.

As said earlier, the problem of multi-membership is solved by choosing the maximum of the newly calculated membership values of a next active state.

The performance of the built fuzzy finite automata was measured by calculating recall, precision and F-measure as follows.

$$F\text{-measure} = 2(\text{recall} * \text{precision}) / (\text{recall} + \text{precision})$$

$$\text{recall} = tp / (tp + fn)$$

$\text{precision} = tp / (tp + fp)$ where tp, fp and fn represent the number of true positives, false positives and false negatives. Fig.3 shows the F-measures obtained for the three representations and the six time slices.

The average F-measure obtained by our method for the raw, change point and last-fired representations are 0.74, 0.76 and 0.78 respectively. There is only slight difference in the performance due to the feature representation methods. The average F-measure values are comparable to the F-measures obtained by the methods described by Kasteren, et al. [23].

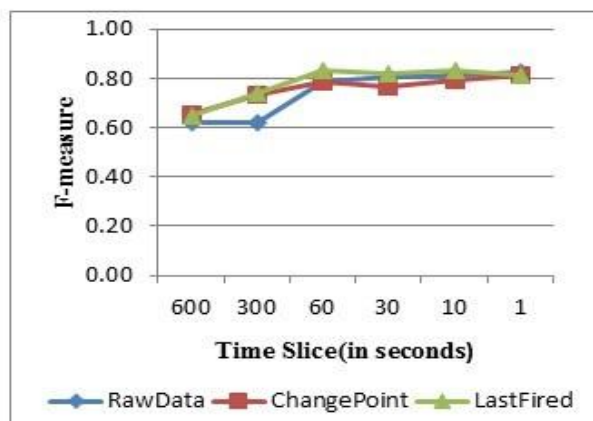


Figure3. F-measure performance

VII. CONCLUSION

Relatively less attention has been paid by researchers to use fuzzy finite automata for activity recognition. In this work we have demonstrated how the automatic construction of a DFA can be used for activity recognition when combined with fuzzy concept. We plan to extend this work by incorporating the ability to recognize temporal characteristics peculiar to the given sensor input sequences.

REFERENCES

- [1] Sanchez D., Tentori M., Favela J. "Activity Recognition for the Smart Hospital", IEEE Intelligent Systems, Volume: 23, Issue: 2, pp. 50 – 57, March-April 2008
- [2] Tim van Kasteren, AthanasiosNoulas, Gwenn Englebienne and Ben Kröse, "Accurate Activity Recognition in a Home Setting", Proceedings of the 10th international conference on Ubiquitous computing, pp. 1-9, 2008
- [3] Christian Wojek, Kai Nickel, Rainer Stiefelhagen, "Activity Recognition and Room-Level Tracking in an Office Environment", IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, pp. 25 – 30, Sept. 2006
- [4] Marco Leo, Paolo pagnolo, TizianaD’Orazio, and ArcangeloDistante, "Human Activity Recognition in Archaeological Sites by Hidden Markov Models", Advances in Multimedia Information Processing - PCM 2004, Lecture Notes in Computer Science Volume 3332, , pp. 1019-1026, 2005
- [5] Weiyao Huang, Jun Zhang, and Zhijing Liu, "Activity Recognition Based on Hidden Markov Models", Proceedings of the 2nd international conference on Knowledge science, engineering and management, pp. 532-537, 2007
- [6] Matthew Brand, Nuria Oliver, and Alex Pentland, "Coupled Hidden Markov Models for complex action recognition", IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’97), pp.994, 1997
- [7] Xucheng Zhang, FazelNaghdy, "Human Motion Recognition through Fuzzy Hidden Markov Model", Proceedings of the 2005 International Conference on Computational Intelligence for Modeling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC’05), pp.450-456, 2005
- [8] SerafeimPerdikis, DimitriosTzovaras, Michael GerasimosStrintzis, "Recognition of human activities using Layered Hidden Markov Models", ZL50 digital Library, <http://en.zl50.com>, 2006
- [9] Md. KamrulHasan, HusneAraRubaiyeat Yong-Koo Lee, Sungyoung Lee, "A Reconfigurable HMM for

- Activity Recognition”, 10th International Conference on Advanced Communication Technology, Volume: 1, Publisher: IEEE, pp. 843-846, 2008.
- [10] RocíoDíaz de León, L. Enrique Sucar, “Recognition of Continuous Activities”, Proceedings of the 8th Ibero-American Conference on AI: Advances in Artificial Intelligence, pp. 875-881, 2002.
- [11] RocíoDíaz de León, “Continuous Activity Recognition with Missing Data”, Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) , Volume 1, page 10439, 2002.
- [12] Emmanuel Munguia Tapia Stephen S. Intille Kent Larson, “Activity Recognition in the Home Using Simple and Ubiquitous Sensors”, Proceedings of Second International Conference on Pervasive Computing, pp.158-175, 2004.
- [13] Sangho Park, J.K. Aggarwal, “Semantic-level Understanding of Human Actions and Interactions using Event Hierarchy”, CVPRW 04, Conference on Computer vision and Pattern Recognition Workshop, June 2004.
- [14] Tim van Kasteren and Ben Kröse, “Bayesian Activity Recognition in Residence for Elders”, 3rd IET International Conference on Intelligent Environments (IE 07), pp. 209 – 212, 2007.
- [15] Justin Muncaster, Yunqian Ma, “Activity Recognition using Dynamic Bayesian Networks with Automatic State Selection”, Proceedings of the IEEE Workshop on Motion and Video Computing, Page 30, 2007.
- [16] Somboon Hongeng, Francois Br émond and RamakantNevatia, “Bayesian Framework for Video Surveillance Application”, Proceedings of the 15th International Conference on Pattern Recognition, 2000.
- [17] Friedrich Steimann and Klaus-Peter Adlassnig, “Clinical Monitoring with Fuzzy Automata”, Journal of Fuzzy Sets and Systems, Volume 61 Issue 1, pp. 37-42, Jan. 10, 1994.
- [18] A. Alvarez-Alvarez, J. M. Alonso, G. Trivino, N. Hernández, F. Herranz, A. Llamazares and M. Ocaña, “Human Activity Recognition applying Computational Intelligence techniques for fusing information related to WiFi positioning and body posture”, 2010 IEEE International Conference on Fuzzy Systems (FUZZ), pp. 1 – 8, 18-23 July 2010.
- [19] Alberto Alvarez-Alvarez, GracianTrivino, Oscar Cordón, “Human Gait Modeling Using a Genetic Fuzzy Finite State Machine”, IEEE T. Fuzzy Systems 20(2), pp.205-223, 2012.
- [20] Gonzalo Bailador, GraciánTriviò, “Pattern recognition using temporal fuzzy automata”, Journal of Fuzzy Sets and Systems, Volume 161 Issue 1, Pages 37-55, January, 2010.
- [21] Henning Fernau, “Algorithms for learning regular expressions from positive data”, Journal of Information and Computation, Volume 207, Issue 4, Pages 521-541, April 2009.
- [22] Carlo A. Furia, “A Survey of Multi-Tape Automata”, Transactions of the IRE professional Group, May 2012.
- [23] T.L.M. van Kasteren, G. Englebienne, and B.J.A. Kröse, “Human Activity Recognition from WirelessSensor Network Data: Benchmark and Software”, Chapter 8, Activity Recognition in Pervasive Intelligent Environments, Atlantis Press, 2011.
- [24] Mansoor Doostfatemeh, Stefan C. Kremer, “New directions in fuzzy automata”, International Journal of Approximate Reasoning, pp. 175–214, 2005.

H Karamath Ali received his M.Sc. and M.Phil. in Computer Science from Bharathidasan University in 1990 and 2001 respectively. He is working as a Associate Professor in Periyar E.V.R. College, Tiruchirappalli and is pursuing his Ph.D. in the area of activity recognition using sensor data in smart environments.

Dr. D I George Amalarethnam received his MCA, MPhil & PhD in Computer Science from Bharathidasan University in 1988, 1996 & 2007 respectively. He is working as an Associate Professor & Director of MCA in Jamal Mohammed College, Tiruchirappalli. He has 25 years of teaching and 15 years of research experience. He has published several papers in International and National Journals and Conferences. He is a Life Member of Indian Institute of Public Administration, New Delhi, Senior Member, International Association of Computer Science and Information Technology, Singapore and Editorial Board Member for many International Journals. His areas of interest are Parallel Processing, Operating Systems and Computer Networks.