International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 3, Issue 5, September-October 2014 **ISSN 2278-6856**

OBCAR : An Algorithm for Object Based Composite Activities Recognition in Smart Environments

H. Karamath Ali[1] and Dr. D. I. George Amalarethinam[2]

[1]Corresponding Author, Associate Professor in Computer Science, Periyar EVR College (Autonomous), Tiruchirappalli, Tamil Nadu, India

[2] Associate Professor & Director-MCA, Jamal Mohamed College (Autonomous), Tiruchirappalli, Tamil Nadu, India

Abstract

Interleaving or concurrently pursuing two or more activities is the nature of human beings. This is more so for activities of daily living(ADLs). So, an activity recognition system must be able to recognize concurrent and interleaved activities in real-life situations without imposing any restrictions on the user. Users must be allowed to freely interleave and switch over activities. In this paper a simple and novel algorithm, named Object Based Composite Activities Recognition(OBCAR) is proposed for recognizing composite activities from environmental sensor data sets. The algorithm makes use of an automatically constructed finite automaton to identify composite activities. The results achieved by the algorithm are highly promising when tested with a publicly available data set.

Keywords:- Pervasive Computing, Context Awareness, Activity Recognition, Concurrency, Interleaving, Finite Automata, Environmental Sensors, Activities Of Daily Living, Smart Environments.

1. INTRODUCTION

Proactivity is essential to offer personalized services in pervasive computing environments [1].A proactive system needs to be aware of the context of the user.Activity pursued by users is an important aspect of context. Therefore ability to recognize human activities unobtrusively is vital for a system to be context aware. An activity recognition system should be able to recognize activities without imposing any restrictions on the user. Users should be allowed to pursue their activities without the need to change their normal routine. It is the nature of people to pursue two or more activities concurrently or by interleaving. It is not possible to predict the way in which users interleave activities. Also the number of activities a user can pursue simultaneouslycannot be restricted. Therefore an activity recognition system must be able to identify the activities irrespective of the number and combination of activities performed by the user. To monitor user activities researchers have used video cameras, on-body sensors such as accelerometers and gyroscopes and sensors embedded in objects handled by the user. Many users may not prefer to be observed by a video camera as their privacy may get compromised. Wearing sensors on different parts of the body may cause inconvenience to the user. Also, the number of ADLs that can be observed using on-body sensors is very limited. So, in this paper, a simple and novel algorithm named Object Based Composite Activities Recognition(OBCAR) is presented to recognize composite activities of daily living(ADLs) using the outputs of sensors embedded in the objects handled by the user. The algorithm uses an automatically constructed automaton and simple two dimensional tables to identify the foreground and background activities performed by the user. The algorithm allows the user to pursue any number of activities and switch over from any activity to any other activity without any restrictions. When tested with a publicly available data set, the algorithm achieves highly promising results. The remainder of this paper is organized as follows: Section 2 presents an overview of the related work in composite activity recognition; section 3 defines the problem statement; section 4 explains the proposed algorithm; section 5 discusses the dataset used and the experiment conducted; and section 6 presents conclusion.

2. RELATED WORK

A two level approach for recognizing multiple goals was proposed by Xiaoyong Chai and Qiang Yang[2]. Using dynamic Bayesian network, the lower level determined the constituent actions of goals by measuring signal strength through a hand held device. Models were instantiated and terminated dynamically in the higher level. Each model was a finite state machine and functioned as a goal recognizer. Multiple-goal behavior was modeled as transitions among some pre-defined states of these models. It was inferred whether one of a user's goals was present or not by distinguishing the state of a model. Derek Hao Hu and Qiang Yang[3] proposed a two-level probabilistic framework – CIGAR (Concurrent and Interleaving Goal and Activity Recognition) - to recognize both concurrent and interleaving goals. Interleaving goals were modeled using Skip-chain Conditional Random Fields (SCCRF) and concurrent goals were modeled by adjusting inferred probabilities through a correlation graph. Marja Ruotsalainen, et al.[4] suggested a Genetic Algorithm based method for Interleaved Sequential pattern detection(GAIS) from event sequences. It was assumed that the models to detect the required kind of patterns from the event sequenceswere available. After generating an initial population of randomly created individuals GAIS calculated fitness value for each individual using models in the model set. Individuals were selected, crossed and mutated based on the fitness value. Experiments for recognition of multiple concurrent activities in the MIT House_n data set were designed by Tsu-yu Wu, et al.[5]. They used Factorial Conditional Random Fields (FCRFs).The data set contained annotated data collected from multiple sensors in a real living environment.

FCRFs were used by Chia-chun Lian, et al.[6] also, to model the conversational dynamics of concurrent chatting behaviors to accommodate co-temporal relationships among multiple activity states. In their observation the Loopy Belief Propagation (LBP) algorithm is inefficient, and they proposed Iterative Classification Algorithm (ICA) as the inference method for FCRFs. For recognizing multitasked activities Joseph Modayil, et al.[7] used Interleaved Hidden Markov Models. The dynamics of both inter and intra activities were captured by the model. That interleaved activities can be recognized by sensors in physical environments was demonstrated by Geetika Singla and Diane J. Cook[8]. They observed that, HMM performed better than Naïve Bayes model. It was demonstrated by Niels Landwehr[9] that an inference algorithm obtained by extending structured approximate inference methods used with factorial hidden Markov models performs better than a standard hidden Markov model in recognizing multiple interleaved activities observed by a stream of sensor outputs. A goal taxonomy that contained several classes of complexity levels and different granularities of activities was defined by Derek Hao Hu, et al.[10], and they related the recognition accuracy with different complexity levels or granularities. They used skip chain CRF for recognizing multiple concurrent and interleaving activities. A method named activity pattern discovery for activity recognition by building a hierarchical activity model was proposed by Eunju Kim, et al.[11]. A supervised learning algorithm was used to recognize the lower-level activities, such as sitting, standing, eating, and driving. Combinations of the lower-level activities representing more complex activity patterns were discovered by the higher level of the model.
The usefulness of Markov logic combined with common-sense background knowledge to develop a framework for recognizing interleaved and concurrent activities was demonstrated by Rim Helaoui, et al.[12] . A logic-based approach using a heuristic search planner to solve the multigoal recognition problem, without the need of plan libraries was presented by Jianxia Chen, et al.[13]. They first proposed the formulation of a multigoal recognition problem based on automated planning. Then, to recognize concurrent and interleaving goals from observed activity sequences, a two level probabilistic plan recognition approach was used. As observed in [14], the models used in the above mentioned works are either computationally very expensive[11], or require manual construction of the domain model initially. The need to devise methods for automatic construction of hierarchy of activities is observed by Derek Hao Hu, et al.[10].

In view of these observations, in this paper a simple and novel method for recognizing concurrent and interleaved activities is proposed. In the proposed method a finite automaton is constructed automatically, as illustrated in [14], to hierarchically represent the different event sequences possible for performing each activity. Then interleaved and concurrent pursuance of activities are identified by simply traversing the paths in this automaton. The proposed algorithm does not need any calculation in the recognition phase. The composite activities are identified by simple table look-up operations. When tested using a publicly available dataset the algorithm gives highly promising recognition rate.

3. The Problem

Theproblem can be defined as follows[14]:A set of activities $\mathbb{G} = \{G_1, G_2, G_3, \dots, G_n\}$ and a set of events $\mathbb{A} = \{A_1, A_2, A_3, \dots, A_m\}$ are given. Each activity is associated with a fixed number of sequences of events in $\mathbb{A}$. An event in $\mathbb{A}$ may appear in more than one event sequence in any of the activities. A sequence of events associated with anactivity represents the events that need to be taken to accomplish the activity. Hence each activity may be accomplished by one or more different sequences of events. The exact number of event sequences possible for anactivity depends upon the application and environment. As has been said in the introduction, a user may pursue multiple activities either in a concurrent or interleaved manner. As the user performs the constituent events of the currently pursued activities, the system has to identify the activities and the corresponding sequenceof events.

4. The Method

As mentioned in Section 3, each activity may be achieved by one or more sequences of events in $\mathbb{A}$. A set $\mathbb{S}$ is formed by collecting each possible event sequence of every activity in $\mathbb{G}$. That is

$$\mathbb{S} = \left\{ A^i_{j1} A^i_{j2} A^i_{j3} \dots A^i_{jk_j}, 1 \le i \le n, 1 \le j \le p_i \text{ and each } A^i_{jk} \in \mathbb{A} \right\}$$

where n is the number of possible activities, p_i is the number of possible event sequences of the i^{th}activity and k_j is the number of events in the j^{th} sequence. A DFA equivalent to $\mathbb{S}$ is constructed using the SL-infer algorithm [15] as explained in [14]. The SL-infer algorithm is extended to store in each state of the DFA, a list of labels that represent the activities corresponding to the event sequences that will lead to the state in a path from the starting state to any one of the final states. So given the current state, it can be easily decided which of the activities share the event sequence that led to the state. Obviously, in each of the final states only one label will get stored. This will indicate which activity is accomplished when the final state is reached.

Since it is necessary to keep track of each of the activities currently being pursued by the user, a table named 'ARTable' is used. There is one row for each activity being pursued, in the 'ARTable'. Each row contains the following information about the corresponding activity.

activity id : a unique identification number assigned for the activity;

state : the state in which the DFA was, when the user switched over to some other activity. It is from this state the DFA has to continue when the user resumes the activity;

finished : this is a boolean value indicating whether the activity is finished or not.

Whenever a new activity is started, a new row is added to the ARTable, with the *activity id*, *state*and *finished* information. The activity pursued at a given time step is called foreground activity. Activities that are started and not being carried out at the present time step are called background activities. Obviously, there will be only one foreground activity. All the remaining will be background activities. At any time step, any of the background activities may become the foreground activity if the user chooses to perform an event belonging to the activity. In that case, the current foreground activity automatically becomes a background activity. To locate the information about the foreground activity, a variable 'currentActivityRow' is used. Whenever the user starts a new activity, details of the activity are appended to 'ARTable' and 'currentActivityRow' is made to point the row. Whenever the user shifts from one activity to another, 'currentActivityRow' is altered to point to the row corresponding to the resumed activity. There is one more table named 'eventCount' of size $n \times$, where is the number of sensors and is the number of observed activities. eventCount[*i*][*j*] is the number of times sensor event *i* occurs in activity *j*. The contents of this table are used to make a decision on the performed activity for a given input event. With this simple setup, the OBCAR algorithm given in Figure 1 is used for recognizing interleaved activities. In the OBCAR algorithm, the variable 'currentState' refers to the state the DFA is in, after the last input event of the foreground activity. δ refers to the transition function of the DFA. For each event input e, the algorithm first checks if there is a transition from the current state of the foreground activity. If so, it means the user is continuing with the foreground activity. Therefore, the 'currentState' and the ARTable entry of the foreground activity are updated and the next event input is awaited. If not, it may be an indication that the user has switched over to some background activity. So, the algorithm checks if there is any background activity for which there is a transition from its last state on the event input e. If such an activity can be found, it is made the foreground activity. The previous foreground activity automatically becomes a background activity. In the case of more than one such background activity, the one which was started earlier is chosen. If no such background activity is available, then the user might have started a new activity. So the OBCAR algorithm checks if there is a transition from the initial state of the DFA on input e. If so, a row with the unique *id* of the started activity and the state reached after the input e, is appended to the ARTable, and the 'currentActivityRow' is made to point the new row. That is, the new activity becomes the foreground activity.

```
get next event e;
if (currentActivityRow != -1 && δ(currentState, e)) is defined then,
        currentSstate = ARTable[currentActivityRow][state] = δ(currentState, e);
else
if there is a background activity, say b, for which δ(state, e) is defined then
        make currentActiviytRow point to row of b;
        currentState = δ(state_b, e);
else
          if δ(intialstate, e) is defined then
                append a row to ARTable with new activity id and δ(initialstate, e);
                make currentActivityRow point to the new row;
                currentState = δ(initialstate, e)
        else
             find the activities in which e appears;
if any of these activities appears in ARTable then
                make the activity foreground activity;
                update currentState and currentActivityRow appropriately;
        else
                select the activity with maximum occurrences of e as the
                        foreground activity;
           end
       end
   end
end
if (currentState is a final state) then
   mark the forground activity 'finished' in ARTable;
   set currentActivityRow = -1, indicating no foreground activity at present;
end
```

Figure 1. Object Based Composite Activities Recognition(OBCAR) Algorithm

Table 1. Observed Activities in Patterson Data Set

1	Clear The Table
2	Eat Breakfast
3	Front Door
4	Make A Vanilla Latte
5	Make Juice
6	Make Oatmeal
7	Make Soft Boiled Eggs
8	Make Tea
9	Set The Table
10	Use The Bathroom
11	Use The Phone

If none of these three conditions hold good, then the algorithm finds the list of activities in which the event e appears. If there is a common activity in the list and the ARTable, then it is made the foreground activity. If there are more than one such common activity, then the one in which e appears most number of times is selected to be the foreground activity. If there is no common activity in the list and the ARTable, then the activity in which e appears most number of times is introduced as the foreground activity.

International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 3, Issue 5, September-October 2014 ISSN 2278-6856

5. Data Set and Experiment

5. 1. Data Set

To test the performance of the OBCAR algorithm, the data set collected by Patterson, et al.[16] was used. It is an established data set used by many researchers. It was used by Helaoui et al.[17] for testing their HMM based models for recognition of interleaved activities. The data set was collected by observing routine morning activities performed using common household objects and are commonly interleaved. Table 1 lists the 11 activities that were observed. Each of the activity was performed 12 times: by itself twice, and then on 10 mornings all of the activities were performed together in a variety of patterns[16]. In order to capture the identity of the objects being

Table 2. Performance of OBCAR in predicting the foreground activity

	True +ves	False +ves / -ves	Total time steps	Precision	Recall	F-measure
Day 01	1375	3	1378	0.998	0.998	0.998
Day 02	2110	37	2147	0.983	0.983	0.983
Day 03	1518	513	2031	0.747	0.747	0.747
Day 04	2165	15	2180	0.993	0.993	0.993
Day 05	2387	51	2438	0.979	0.979	0.979
Day 06	2652	9	2661	0.997	0.997	0.997
Day 07	2357	26	2383	0.989	0.989	0.989
Day 08	2458	225	2683	0.916	0.916	0.916
Day 09	2560	52	2612	0.980	0.980	0.980
Day 10	2611	14	2625	0.995	0.995	0.995

manipulated, the kitchen was outfitted with 60 Radio Frequency Identification(RFID) tags placed onevery object touched by the user during a practice trial. The user simultaneously wore two gloves, outfitted with antennae that were able to detect when an RFID tag was within 2 inches of the palm. The time and identification code of every object touched was sent wirelessly to a database for analysis. The activities were not performed sequentially or in isolation. During a pause in any given activity, progress was attempted in other parallel activities (such as when waiting for water to boil) and some activities interrupted others at uncontrolled times (such as answering the phone). The dataset comprises two subsets: ‘‘standard data’’ and ‘‘full data’’. The ‘‘standard data’’ is provided in the form of timely ordered events relating, for each time step, the ID of active sensors and the activity being carried out. Unlike the standard data, the full data provides all concurrent activities for each time step.

5. 2. Experiment

The OBCAR algorithm was evaluated with two different recognition tasks:

1. Predicting the foreground activity for each event.
2. Deriving all background activities at each event.

Performance of the algorithm in each of these tasks was measured by calculating precision, recall and F-measure values For predicting the foreground activity for each event, correctly predicted activities are counted as true positives (TP). A predicted activity ‘a1’ at time step ‘t’ that does not match the activity ‘a2’ in the reference dataset for the same time step is counted as a false positive (FP). In addition, the activity ‘a2’ counts as false negative (FN) since it is present in the reference dataset but missing in the prediction for the same time step. It must be noted that the number of false positives and false negatives coincide. This is due to the fact that at each time step only one activity occurs[17]. The results obtained by the algorithm using 10 fold cross validation are given in Table 2 and in Figure 2. Since the number of false positives and false negatives coincide, the precision, recall and F-measure values remain

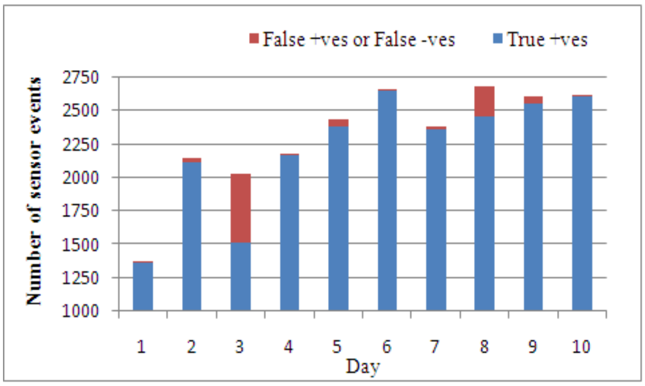


Figure 2. Performance of OBCAR in predicting the foreground activity

the same. The average F-measure value obtained is 0.96. This is more than the F-measure values 0.92 and 0.93 obtained by Helaoui et al.[17] for the same data set. Task 2 is about deriving all background activities at each time step. In other words, it is about testing the ability of the algorithm to identify the interleaved activities. Recognition of Interleaved activities is evaluated as follows:at each sensor event the set of predicted activities is compared to all activities in progress(reference set). Eachpredicted activity that is absent in the reference set is counted as a false positive. Each activity missing in the prediction set but present in the reference set is counted as false negative. Correctly predicted activities are true positives. Finally, if an activity is not in progress and has not been predicted it counts as true negative. Then precision, recall and F-measure values are calculated as usual. The values obtained by the OBCAR algorithm using 10-fold cross validation are tabulated in Table 3 and illustrated in Figure 3. In Table 3, total count value for a day was obtained by multiplying the number of events in the data set for the day by 11. This is because, for each event input, for each of the 11 activities, true/false positive/negative value needs to be decided based on the occurrence of the activity in the reference and predicted sets.

Table 3. Performance of the OBCAR algorithm in Recognizing Interleaved Activities

	Total count	True +ves	False +ves	True -ves	False - ves	Precision	Recall	F-measure
Day 01	15158	2748	3	12403	4	1.00	1.00	1.00
Day 02	23617	4040	51	19385	141	0.99	0.97	0.98
Day 03	22341	3320	500	18008	513	0.87	0.87	0.87
Day 04	23980	3908	16	20037	19	1.00	1.00	1.00
Day 05	26818	4810	1028	20979	1	0.82	1.00	0.90
Day 06	29271	5443	10	23808	10	1.00	1.00	1.00
Day 07	26213	4217	169	21804	23	0.96	0.99	0.98
Day 08	29513	5375	558	23566	14	0.91	1.00	0.95
Day 09	28732	4803	2075	21853	1	0.70	1.00	0.82
Day 10	28875	4726	2726	21306	117	0.63	0.98	0.77
					Average	0.89	0.98	0.93

The precision, recall and F-measure values obtained by Helaoui et al.[17], are 0.73, 0.99 and 0.84 respectively. The corresponding values achieved by the OBCAR algorithm are 0.89, 0.98 and 0.93 respectively. This is shown in Figure 4. While the OBCAR algorithm achieves superior value for precision, there is a slight dip in recall. The F-measure value of the OBCAR algorithm is much better than that of Helaoui et al.[17].

6. CONCLUSION

It is important for an activity recognition system to be able to recognize concurrent or interleaved multiple activities performed by users in a smart environment. To this end, in this paper a simple and novel algorithm named Object Based Composite Activities Recognition Algorithm is proposed. By monitoring the user's interaction with objects, OBCARalgorithm recognizes compositeactivities pursued by a user. OBCAR algorithm does not impose any restriction on the number of activities that can be interleaved. The user is allowed to pursue any number of activities as he/she wishes.

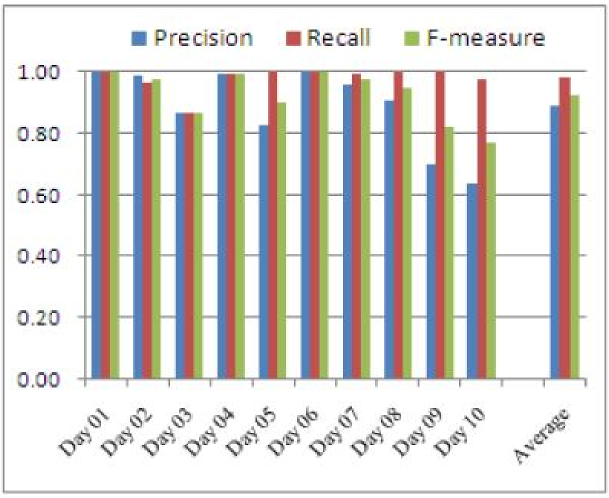


Figure 3. Performance of the OBCAR algorithm in recognizing interleaved activities

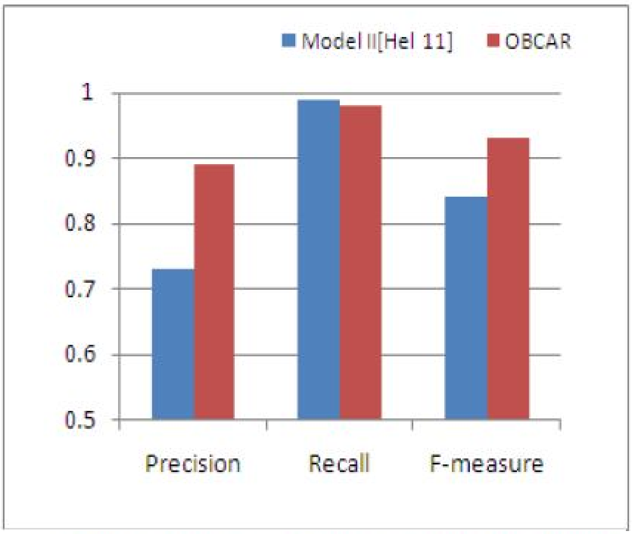


Figure 4. Performance Comparison for Recognition of Interleaved Activities

The user is allowed to switch over from any activity to any other activity. Also, no calculations are necessary during recognition of the activities. Simple table look-up operations are all that is necessary for identifying the Figure 3. Performance of the OBCAR algorithm in recognizing interleaved activities Figure 4. Performance Comparison for Recognition of Interleaved Activities foreground and background activities. It is shown that the algorithm achieves highly promising results when tested with an established and publicly available data set

REFERENCES

[1] M. Satyanarayanan, "Pervasive Computing: Vision and Challenges", IEEE Personal Communications, August 2001, pp. 10-17.

[2] Xiaoyong Chai and Qiang Yang, "Multiple-Goal Recognition from Low-Level Signals", American Association for Artificial Intelligence, 2005, pp. 3-8.

[3] Derek Hao Hu and Qiang Yang, "CIGAR: Concurrent and Interleaving Goal and Activity Recognition", Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, 2008, pp. 1363-1368.

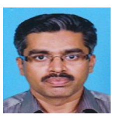
[4] Marja Ruotsalainen, Timo Ala-Kleemola and Ari Visa, "GAIS: A Method for Detecting Interleaved Sequential Patterns from Imperfect Data", Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2007), 2007, pp. 530-535.

[5] Tsu-yu Wu and Chia-chun Lian and Jane Yung-jen Hsu, "Joint Recognition of Multiple Concurrent Activities using Factorial Conditional Random Fields", Proceedings of the 22nd Conference on Artificial Intelligence, (AAAI), 2007.

[6] Chia-chun Lian and Jane Yung-jen Hsu, "Probabilistic Models for Concurrent Chatting Activity Recognition", ACM Transactions on Intelligent Systems and Technology (TIST), Volume 2 Issue 1, January 2011,Article No. 4, pp. 1138-1143.

[7] Joseph Modayil, Tongxin Bai and Henry Kautz, "Improving the Recognition of Interleaved Activities", UbiComp'08, September 21-24, 2008, pp. 40-43.
[8] Geetika Singla and Diane J. Cook, "Interleaved Activity Recognition for Smart Home residents", Proceedings of the fifthInternational Conference on Intelligent Environments, 2009.
[9] Niels Landwehr, "Modeling Interleaved Hidden Processes", Proceedings of the 25th International Conference on Machine Learning, 2008.
[10] Derek Hao Hu, Sinno Jialin Pan, VincentWenchen Zheng, Nathan Nan Liu and Qiang Yang, "Real World Activity Recognition with Multiple Goals", UbiComp'08, September 21-24, 2008.
[11] Eunju Kim, Sumi Helal and Diane Cook, "Human Activity Recognition and Pattern Discovery", IEEE Pervasive Computing, Volume 9, issue 1, 2010, pp. 48-53.
[12] Rim Helaoui, Mathias Niepert and Heiner Stuckenschmidt, "Recognizing Interleaved and Concurrent Activities: A Statistical-Relational Approach", Proceedings of IEEE International Conference on Pervasive Computing and Communications, 2011, pp. 1-9.
[13] Jianxia Chen, Yixin Chen, You Xu, Ruoyun Huang, Zheng Chen, "A Planning Approach to the Recognition of Multiple Goals", International Journal Of Intelligent Systems, vol. 28, 2013, pp. 203–216.
[14] H. Karamath Ali and D. I. George Amalarethinam, "Recognition of Concurrent and Interleaved Activities in a Smart Environment using Finite Automata", International Journal of Computer Applications,Volume 88 – No.16, February 2014, pp. 41-45.
[15] Henning Fernau, "Algorithms for learning regular expressions from positive data", Journal of Information and Computation, Volume 207, Issue 4, April 2009, pp. 521-541.
[16] Patterson DJ., Fox D., Kautz H. and Philipose, "Fine-grained activity recognition by aggregating abstract object usage", Proceedings of IEEE International Symposium on Wearable Computers, 2005, pp. 44–51.
[17] Rim Helaoui, Mathias Niepert and Heiner Stuckenschmidt, "Recognizing interleaved and concurrent activities using qualitative and quantitative temporal relationships", In Pervasive and Mobile Computing, Elsevier, 2011, pp. 660-670.

AUTHOR



H. Karamath Alireceived his M.Sc. and M.Phil. in Computer Science from Bharathidasan University in 1990 and 2001 respectively. He is working as an 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 Amalarethinam 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 26 years of teaching and 16 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