Progress in Nonlinear Dynamics and Chaos
Vol. 2, No. 1, 2014, 23-33
ISSN: 2321 – 9238 *(online)*
Published on 24 December 2014
www.researchmathsci.org

Progress in Nonlinear Dynamics and Chaos

Application of Fuzzy System Rules to Human Activity

***M. Kiruthiga*[1] and *C. Loganathan*[2]**

[1]Department of Mathematics, Maharaja Arts and Science College, Coimbatore, India
Corresponding author. Email: Kiruthi.neha@gmail.com
[2]Principal, Maharaja Arts and Science College, Coimbatore, India
Email: clogu@rediffmail.com

Received 13 November 2014; accepted 12 december 2014

Abstract. Intelligent systems are developed to explain real time situations and to model them from mathematical point of view. Since these systems do not obey the principle of super possession and homogeneity, they are not linear. These systems therefore have nonlinear dynamical behavior. Along with that, these systems undergo uncertainty and there is an amount of probability involved in their behavior. We developed a fuzzy system based on fuzzy intervals and fuzzy optimization process. The results are illustrated from standard situations in modeling human activities.

***Keywords**: fuzzy system, fuzzy optimization, fuzzy intervals, human activities, intelligent systems*

1. Introduction

Schalkoff has developed principles, paradigms to study intelligent systems. In that, he has modeled Fuzzy Intelligent System using based on control system results on linguistic variables [13].

Description-based approaches are a particular class of hierarchical recognition methodologies designed for analysis of high-level activities [2,5,6,7,13]. The motivation behind description-based approaches is to recognize human activities by maintaining the activities' temporal structure. Using time intervals and temporal predicates [7] to represent the structure of each activity, previous approaches have obtained successful results on recognizing high-level human activities, by searching for visual inputs that satisfies the activities' structure. Description-based approaches are able to overcome the limitations of previous statistical and syntactic approaches [3,4] on recognizing concurrently organized activities. We present a reliable human activity recognition methodology which handles the structural variations of an activity. When a new observation (i.e. video) containing an execution of an activity is provided, our system measures how semantically similar a given observation is to the optimal structure of the activity. This similarity measure is not deterministic but is designed to consider uncertainties of the activities' structures.

Over all process of the system is as follows. At each occurrence of gestures, we associate a fuzzy [9, 10] time interval. In contrast to a deterministic time interval used by previous approaches [2, 6,7,8], a fuzzy interval is able to describe a possible range of its starting time and that of its ending time as well as the confidence value associated with time frames within the ranges. Once fuzzy intervals are calculated, a dynamic programming algorithm that we have designed and presented here is applied to measure the similarity between the detected fuzzy intervals and the structure of the activity specified in the representation. Our algorithm searches for the time points in ranges that satisfy the temporal structure specified in the activity representation while maximizing the fuzzy membership values. A logistic regression technique has been used to estimate the similarity function.

2. Basic concepts and results

2.1. Fundamental fuzzy system concepts include

1. The notion and quantification of fuzzy sets, especially the all-important notion of membership functions for (fuzzy and non fuzzy. i.e., crisp) sets;
2. Linguistic variables, labels, and hedges;
3. The process of fuzzification;
4. The propagation of fuzzy information via fuzzy productions (rules) and associated compositional rules of inference (CRI); and
5. The process of defuzzification.

2.2. Determining membership functions, μi

There are a number of ways to acquire the necessary μi for a fuzzy system. These include:

1. Subjective evaluation and elicitation (experts specify membership function curves appropriate to a given problem).
2. Ad-hoc functional forms (most actual fuzzy control operations draw from a very small set of different curves).
3. Converted frequencies or probabilities. (However, we must remember that membership functions are NOT probabilities.)
4. Physical measurement.
5. Learning and adaptation.

2.3. General fuzzy system structure

1. Major components

A general structure for a system employing fuzzy concepts consists of three entities:

- A fuzzification process that converts non fuzzy (crisp) inputs into their fuzzy counterparts;
- A fuzzy computational mechanism (CRI) that maps fuzzy quantities into fuzzy quantities; and
- A defuzzification interface that converts the fuzzy-domain results into nonfuzzy (crisp) outputs.

Not all three components are required. Only the heart of the system shown, i.e., the fuzzy computational mechanism, is required. In the case of a system based upon linguistic (fuzzy) inputs, fuzzification is not required. Similarly, if a linguistic (fuzzy set) output is sufficient, defuzzification is not required.

2. Linguistic variables

Perhaps the most fundamental element in fuzzy systems is the notion of a linguistic variable, i.e., a variable whose values are words rather than numbers. For example, consider the use of the linguistic variable temperature in the fuzzy expression: "the temperature is hot," as compared with the crisp interpretation: "the temperature is 56.00108709023degreeC." Linguistic variables also allow qualifiers on the fuzzy set or linguistic label/descriptor, for example, valid expressions are " the temperature is very hot," " the temperature is NOT hot" and " the temperature is NOT very hot."

3. Fuzzy Antecedents and Rules

The structure of an antecedent in a fuzzy rule is:

u is the V $\Longrightarrow$ "input temperature is hot"

This leads to the rule form4:

IF (x is A) and (y is B) THEN z is C

where

♦ A and B denote fuzzy sets over the input domains X and Y of linguistic variables x and y.

♦ C is a fuzzy set over the output domain Z of linguistic variable z.

3. Fuzzy optimization

The precise quantification of many system performance criteria and parameter and decision variables is not always possible, nor is it always necessary. When the values of variables cannot be precisely specified, they are said to be uncertain or fuzzy. If the values are uncertain, probability distributions may be used to quantify them. Alternatively, if they are best described by qualitative adjectives, such as dry or wet, hot or cold, clean or dirty, and high or low, fuzzy membership functions can be used to quantify them. Both probability distributions and fuzzy membership functions of these uncertain or qualitative variables can be included in quantitative optimization models. This chapter introduces fuzzy optimization modeling, again for the preliminary screening of alternative water resources plans and management policies.

4. Fuzzy time intervals

In this section, we introduce the concept of ‘fuzzy Time intervals’, which is designed to capture uncertainties and variations in activity executions. A time interval is a pair of starting time and ending time, which describes the time associated with an occurring activity or sub-event. Previously, most of approaches have used deterministic time intervals with strictly fixed starting and ending time to describe detected sub-events, and analyzed their relationships to recognize activities [8, 2, 6, 7]. Our system associates ‘fuzzy intervals’ for detected actions; we adopt the concept of fuzzy sets, and describes each starting time or ending time of actions as a fuzzy range of time rather than a time point. Each frame (i.e. discrete time points) within the range will have corresponding fuzzy set function value describing how confident the system is on the fuzzy interval. Fuzzy intervals are not only associated with atomic actions, but also associated with high-level actions and interactions for the hierarchical recognition.

Fuzzy intervals and possible time intervals described by fuzzy intervals. Two fuzzy intervals whose ranges overlap slightly. Possible time intervals extracted from the fuzzy intervals, which are selected to satisfy the temporal relationship meets. A starting or ending point of each time interval has an associated fuzzy set value describing how confident the system is on the time interval.

Clearly illustrates why fuzzy time intervals are more desirable than traditional deterministic intervals on handling variations of an activity. Assume that a structure of an activity is described as time intervals of two sub-events occurring in a sequence (i.e. meets). When two overlapped intervals are detected (instead of sequential ones) due to an execution variation, systems using the deterministic time intervals whose starting times and ending times are fixed to local maximums fail to recognize the activity. On the other hand, as illustrated, the fuzzy intervals contain time intervals that meets with a certain confidence, thereby enabling the recognition.

A fuzzy interval describes a set of possible time intervals. Among sets of possible time intervals described by fuzzy intervals of sub-events, there may exist particular choices that make the temporal constraints of an activity to be satisfied. The goal is to make the system search for such selections while maximizing fuzzy values associated with intervals, so that the overall similarity using fuzzy intervals can be measured. We present the detailed algorithm to measure such similarity in the following section.

In principle, our recognition methodology is able to cope with any function as a fuzzy membership function associated with starting or ending point of a time interval. We have chosen a triangular function that is commonly used in fuzzy logic to be the fuzzy function of starting or ending point of an atomic-level action. Based on the training data, a variance of a starting or ending time of each atomic actions has been measured, and the height and the width of the triangle function have been empirically decided. The fuzzy function of higher-level activities are calculated hierarchically, as a consequence of the recognition process.

5. Recognition algorithm

In this section, we present an algorithm to recognize human activities using fuzzy intervals. We first present a methodology to recognize activities by measuring similarities between its structure and observations (i.e. detected fuzzy intervals of sub-events). A hierarchical similarity measurement is presented next.

The problem of recognizing an activity based on detection results of sub-events can be formulated as follows: Given fuzzy intervals associated with each subevent, the goal is to search for a valid combination of time intervals within the ranges of fuzzy intervals that maximize the fuzzy values (i.e. confidence) while satisfying the temporal constraints of the activity. If the assigned fuzzy values are high enough, the system is able to deduce that given fuzzy intervals are similar to the activity's structure and conclude that the activity occurred. In order to integrate fuzzy values associated

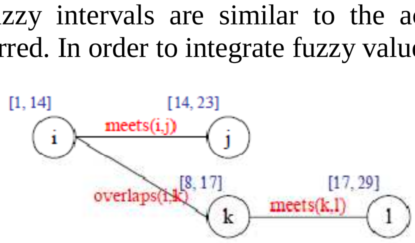

Figure 1. An example temporal graph

with sub-events to calculate the overall confidence of the occurring activity formally, we have used a logistic regression technique. Confidence of the activity is computed as a weighted sum of starting and ending times' fuzzy values fitted into the logistic function. Let ($v_1, \ldots, v_n$) be time intervals within the ranges of fuzzy intervals of n sub-events, and ($x_1, \ldots, x_n$) be their fuzzy membership values. Then, overall fuzzy confidence of the activity, L, is measured as

$$L = max(L(x_1, \ldots, x_n)) = logit^{-}1(max(F(x_1, \ldots, x_n))) \quad (1)$$

where

$$F(x_1, \ldots, x_n) = b + a_1 \_ (x^s{}_1 + x^e{}_1) + \ldots + a_n \_ (x^s{}_n + x^e{}_n) \quad (2)$$

where $x^s{}_k$ indicates the fuzzy value of the starting time of v_k and $x^e{}_k$ indicates that of ending time. The function *logit* is defined as $logit(p) = ln(p/(1 - p))$, and $a_1, ..., a_n$ and b are constant weight values which need to be trained.

The system is required to maximize the $F(x_1 , ..., x_n)$ function while meeting temporal constraints posed for ($v_1, \ldots, v_n$). There exist various temporal constraints that the time intervals have to satisfy depending on the representation of the activity. Most trivial constraint is that the starting time of a subevent can not exceed its ending time. Representation of the activity also specifies other constraints using temporal predicates. For example, if the representation contains *before*(v_1, v_2), then the ending time of v_1 must be strictly less than the starting time of v_2. Choosing time point 9 for $v^e{}_1$ and 5 for $v^s{}_2$ leads to a contradiction, regardless their fuzzy values.

In order to compute $max(F(x_1, \ldots, x_n))$ while satisfying the constraints, we have developed a dynamic programming algorithm. We first convert temporal representation of an activity into an undirected acyclic graph representation (i.e. tree) where each node corresponds to a time interval and each edge specifies that two intervals are required to satisfy a particular relationship. An edge is labeled with the relationship that needs to be satisfied between the two nodes (e.g. $during(v_1, v_2)$). Multiple graphs may be constructed from disjunctive normal form (DNF) of the representation. Figure 1 shows an example temporal graph of the interaction 'push' mentioned.

We formulate the recursive equation as:

$$G_k(t) = a_k \,.\, (x^s{}_k + x^e{}_k) + \sum_{all\ v_c} max\{t'\}G_c(t') \quad (3)$$

where v_c are child nodes of v_k, and t' are time intervals that satisfies temporal relations with the interval t. $G_k(t)$ specifies the maximum weighted sum of possible assignments for x_k and its descendant nodes, if the interval t is assigned for x_k. Therefore, the similarity measure $L(x_1, x_2, \ldots, x_n)$ are enumerated as follows:

$$\begin{aligned} max(L(x_1, \ldots, x_n)) &= logit^{-}1\,(max(F(x_1, \ldots, x_n))) \\ &= logit^{-}1(max\{t\}G_r(t)) \end{aligned} \quad (4)$$

where node v_r is the root node of the tree.

As a result, by solving the recursive equation using the dynamic programming algorithm, we are able to calculate the maximum L, which is the confidence of the detection. Furthermore, we are able to calculate the fuzzy interval associated with the detection. By calculating the argument maximum while computing the maximum, we also are able to compute the exact time intervals of sub-events that make the fuzzy value of the activity to be the maximum. This implies that the system is able to calculate the starting time and ending time of the special time interval 'this', which is always associated with the defining activity itself. Ranges are associated with the detected starting and ending time of 'this', making the interval to be fuzzy. The overall complexity of the algorithm is $O(m^2)$, where m is the average number of intervals within ranges per node.

We have developed a hierarchical algorithm which analyzes human activities from bottom (i.e. atomic-level actions) to top (i.e. high-level interactions). At the bottom level, the system detects atomic-level actions (e.g. arm stretching) using low-level recognition techniques such as hidden Markov models (HMMs) from Park and Aggarwal [8], and associates fuzzy intervals to describe their starting and ending time. Higher-level activities are recognized based on fuzzy intervals associated with their sub-events, which are atomic-level actions and/or other activities composed of their own subevents. With the fuzzy interval calculation method presented above in this section, fuzzy intervals of an activity are computed based on those of sub-events, enabling the recognition of high-level activities.

6. Experiments and results

We have evaluated the performance of our system using fuzzy time intervals, while comparing it with the previous systems [10, 11] using deterministic intervals. Eight types of relatively simple interactions between humans (*approach, depart, point, shake-hands, hug, punch, kick, and push*), as well as complex recursive interactions of *fighting* and *greeting* have been tested by the systems. We have used the dataset used in [10, 11], which contains sequences of continuous executions of activities in 320*240 resolutions at 15 fps. Complex fighting-related sequences containing a total of 53 simple and recursive activities have been newly added. As a result, a total of 161 activity executions have been tested for both systems. HMMs for gesture recognition and logistic regression weights, $a_1,\ldots, a_n$ and b, have been estimated based on a separate training set.

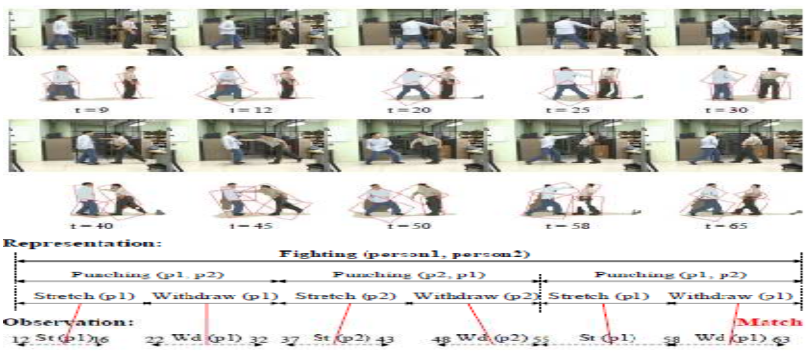

Figure 2. Example experimental results of the recursive activity 'fighting', composed of three consecutive 'punching' interactions.

The experimental results clearly illustrate that the recognition accuracy of our system is better than that of the previous system. Table 1 compares true positive rates obtained from two systems whose false positive rates are similar. The result confirms that the use of fuzzy time intervals helps reliable recognition of activities from noisy videos with structural execution variations. Figure 2 shows a successful recognition result of our system tested on a *fighting* interaction composed of three punching interactions, which the previous deterministic systems failed recognition due to its structural variation. Figure 3 shows a recognition result of *pushing*. Even though the structure of the gesture recognition results was slightly different from the representation, our system was able to recognize the *pushing* interaction. False positive rates were almost 0 for both systems, since the probability of sub-events satisfying particular relations detected 'by accident' is extremely low.

System	Simple	Recursive	Total
Ours	0.920	0.783	0.907
Previous	0.862	0.522	0.814

Table 1. Recognition accuracy

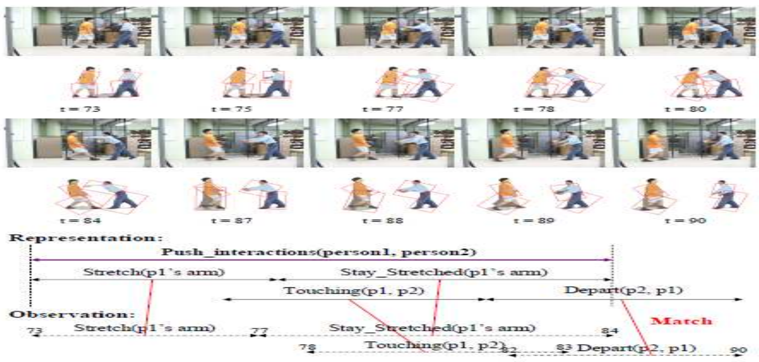

Figure 3. Example experimental results of the 'pushing' interaction

7. Representation

The representation for composite actions must consist of two parts: a list of variables corresponding to time intervals associated with designated sub-events, and the relationships among those variables. The first component can be represented by associating one symbol name with one sub-event. The second component, which represents necessary conditions for composite actions, is defined through predicates mentioned. Variables defined and the special variable 'this', representing defining action itself, are used in order to specify the relationships. Therefore, we are able to represent a composite action in terms of the relationship between 'this' and other time interval variables 't1', 't2', ..., which are satisfying time intervals of sub-events.

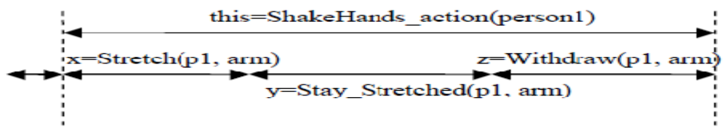

Figure 4: Example illustrating the atomic actions' time intervals and their relationships needed for the composite action, 'shake-hands action'.

As a format of the representation scheme, we use a context-free grammar (CFG). CFG naturally leads the representation to use concepts recursively, enabling the action to be defined based on sub-events. In our representation, atomic actions serve as terminals. On the other hand, composite actions are treated as non-terminals. These non-terminals can be converted to terminals recursively, using production rules.

Our CFG does not generate sequences of poses or gestures directly. Rather, we construct a representation of composite actions using the CFG. A representation built through the CFG describes all participating sub-events, and their relationships. Sub-events can either be atomic actions or other already represented composite actions. Even though the CFG does not create the sequences of poses or gestures directly, we will be able to recognize composite actions through detecting sequences that satisfy the representation constructed with our CFG. With our CFG, we are able to represent any actions if their relationship can be described in terms of the predicates we have defined.

Therefore, the general representation of composite actions can be described using the following context-free-grammar. Non-terminal *Action(i)* indicates action of person i. *Action(i)* can be either an atomic action, or a composite action defined with two components: *Action Defs(i,var)* and *Action Relationship(var)*. The first component, *Action Defs(i,var)*, defines the variables for corresponding time intervals of sub-events. Parameter *var* is defined to be the list of variables associated with sub-events. *Action Defs (i,var)* is the list of several *def(c, Action (i))*, and this defines the contents of list *var*. Statement *def(c, Action (i))* associates some variable c with the time interval of a denoted sub-event. As a result, list *var* contains a list of variables associated with time intervals of corresponding composing events.The second component is *Action Relationship(var)*. With temporal and logical predicates, *Action Relationship(var)* defines the all necessary conditions for the action using all variables in *var* and special variable 'this'. A combination of any temporal predicates presented can be used to define *ActionRelationship(var)*. The time interval 'this' satisfying all necessary conditions will be the corresponding time interval for the action.

Action(i)
    ->(Action Defs(i,var), Action Relationship(var))
    -> atomic _action(operation triplet)
Action Defs (i, var)
    -> list(def(c, Action(i)), Action Defs (i, var-c))
    -> def(c, Action(i))
Action Relationship(var)
    -> Logical-Predicate(Action Relationship(var),
Action Relationship(var))
    -> Temporal-Predicate('this', var(a))
    -> Temporal-Predicate(var(a), var(b))

For example, let's look into the composite action 'shake hands action' again. As we informally defined previously in Figure 4, we associate variable 'x', 'y', and 'z' with sub-events 'Stretch', 'Stay_ Stretched', and 'Withdraw'. Then, relationships are represented in terms of predicates: *meets(x, y)*, *meets(y, z)*, *starts(x, this)*, and *finishes(z, this)*. Therefore, formal representation of 'shake-hands action' is defined through our CFG scheme as follows.

Stretch _hand(i) = atomic _action(<person I 's hand, stretch, other person>)
Stay _Stretched _hand(i) = atomi c _action (<person I 's hand, stay stretched, other person's hand>)
Withdraw _hand(i) = atomic _action(<person i's hand, withdraw, null>)
SH Action Def s (i, var) = list(def('x', Stretch _hand(i)),
list(def('y', Stay_ Stretched _hand(i)),
def('z', Withdraw _hand(i))))
SH Action Relationship (var) = and(meets('x', 'y'), and(
meets('y', 'z'), and(starts('x', 'this'), finishes('z', 'this'))))
Shake Hands _action(i) = (SH Action Defs (i, var), SH Action Relationship(var))

8. Experimental results

We recognized the following eight two-person interactions through our system: approach, depart, point, shake-hands, hug, punch, kick, and push. Interaction videos taken by Sony Handy Cam were converted into sequences of image frames with 320*240 pixel resolution, obtained at a rate of 15 frames per sec. Six pairs of persons participated in the experiment and 24 sequences were obtained. In each sequence, participants were asked to perform a number of above interactions consecutively and continuously. Overall, each interaction was performed 12 times total throughout all sequences.

The representations for the eight interactions were constructed manually using our CFG-based representation scheme. Usually, a composite action is first defined in order to represent meaningful one-person movement in the interaction. For example, in the previous sections, the composite action 'shake-hands action' was defined first in order to represent interaction 'shake-hands interaction'. The composite action 'shake-hands action' and the interaction 'touching' were sub-events.

Figure 3 and 4 show the intermediate outputs of each layer. In this experiment, two persons performed three interactions consecutively: shake-hands, point, and hug. The body-part layer extracts features for each body parts per frame. Figure 3 shows the sequences of raw images, and processed images for extracting body-part parameters. Once the features for each frame are extracted, the pose layer converts them into discrete pose for each body part. The gesture layer converts sequences of poses into sequences of gestures. The recognition algorithm provided then used to recognize interactions based on information from the gesture layer. Figure 4 shows the result of the pose layer, the gesture layer, and the final result of interaction recognitions.

Table 1 shows the performance of our recognition system. Because of the accurate representation on composite actions, the system is superior to all previous systems. Moreover, the results are obtained from sequences of consecutive interactions, not segmented manually. The system was able to recognize sequences of actions and interactions with high degree of accuracy.

9. Conclusion

We have presented a reliable recognition methodology that is able to handle uncertainties in human activities' structure. We have introduced the concept of 'fuzzy time intervals', and presented the dynamic programming algorithm to calculate the similarity between the activity and the observations. Experimental results suggest that the ability to handle structural variations enables better recognition of human activities.

We presented the general methodology for automated recognition of complex human actions and interactions. The fundamental idea is to use the CFG-based representation scheme to represent composite actions and interactions. The CFG-based representation scheme provides a formal method to define occurring time intervals of composite actions and interactions. The idea of representing complex actions and interactions as a composition of simpler actions and interactions was the key. Our experiments show that the system can represent and recognize composite actions and interactions with high recognition rate.

The novelty of our work is on the framework to represent and recognize high-level hierarchical actions from raw image sequence. Our representation explicitly captures the hierarchical nature of actions and interactions. Our system has the ability to use represented actions as sub-events of higher-level actions, thereby minimizing the redundancy. The potential of our work is that our system is able to recognize even higher-level composite actions and interactions. Our system can recognize any actions and interactions if their time intervals can be defined properly through our CFG-based representation scheme. Our framework is also able to handle noisy inputs through HMMs. However, current framework cannot process large scale errors, such as insertion or deletion of sub-events. In the future, we plan to take probabilistic nature of actions into consideration. Also, we aim to develop methodology for our system to learn activity representations based on large training sets.

Interaction	Total	Correct	Accuracy
Approach	12	12	1.000
Depart	12	112	1.000
Point	12	11	0.917
Shake hands	12	11	0.917
Hug	12	10	0.833
Punch	12	11	0.917
Kick	12	10	0.833
Push	12	11	0.917
Total	96	88	0.917

Table 2: Recognition accuracy of the system

REFERENCES

1. J. F. Allen and G. Ferguson, Actions and events in interval temporal logic, *Journal of Logic and Computation*, 4(5) (1994) 531–579.
2. S.Hongeng, R.Nevatia and F.Bremond, Video-based event recognition: activity representation and probabilistic recognition methods, *CVIU*, 96(2) (2004) 129–162.
3. Y.A.Ivanov and A.F.Bobick, Recognition of visual activities and interactions by stochastic parsing, *IEEE T PAMI*, 22(8) (2000) 852–872.
4. N.T.Nguyen, D.Q.Phung, S.Venkatesh and H.H.Bui, Learning and detecting activities from movement trajectories using the hierarchical hidden markov models, *In CVPR* (2) (2005) 955–960.
5. C. S. Pinhanez and A. F. Bobick. Human action detection using PNF propagation of temporal constraints, *In CVPR*, (1998) 898.

6. M. S. Ryoo and J. K. Aggarwal, Recognition of composite human activities through context-free grammar based representation, *In CVPR* (2) (2006) 1709–1718.
7. M. S. Ryoo and J. K. Aggarwal, Semantic understanding of continued and recursive human activities, *In ICPR*, (2006) 379–382.
8. V.-T.Vu, F. Br´emond and M.Thonnat, Automatic video interpretation: A novel algorithm for temporal scenario recognition, *In IJCAI* (2003) 1295–1302.
9. L.A.Zadeh, Fuzzy sets, *Information and Control*, 8(3) (1965) 338–353.
10. L.A.Zadeh, Toward human level machine intelligence - is it achievable? the need for paradigm shift, *IEEE Computational Intelligence Magazine*, Augest (2008) 11–22.
11. M.Kiruthiga and C.Loganathan, Software reliability modeling in fuzzy environment, *Progress in nonlinear Dynamics and Chaos*, 2 (2014),10-17.
12. M.Kiruthiga and C. Loganathan, A refinement on the earlier techniques for nonlinear programming problem in fuzzy environment, *Progress in nonlinear Dynamics and Chaos*, 1 (2013) 54-60.
13. R.J. Schalkoff, Intelligent systems principles, paradigms, and pragmatics, Jones and Bartlett India Pvt Ltd, New Delhi, 2011.