Action recognition in video sequences using a Mealy machine

L. Rodriguez-Benitez, J. Moreno-Garcia, J.J. Castro-Schez, C. Solana and, L. Jimenez

Abstract—In this paper the use of sequential machines for recognizing actions taken by the objects detected by a general tracking algorithm is proposed. The system may deal with the uncertainty inherent in medium-level vision data. For this purpose, fuzzification of input data is performed. Besides, this transformation allows to manage data independently of the tracking application selected and enables adding characteristics of the analyzed scenario. The representation of actions by means of an automaton and the generation of the input symbols for finite automaton depending on the object and action compared are described. The output of the comparison process between an object and an action is a numerical value that represents the membership of the object to the action. This value is computed depending on how similar the object and the action are. The work concludes with the application of the proposed technique to identify the behavior of vehicles in road traffic scenes.

Keywords—approximate reasoning, finite state machines, video analysis.

I. Introduction

In recent years there has been a considerable growth in the development of a great number of systems for security, home automation, zone traffic regulation, etc... that are based on automatic video analysis. A new approach to the recognition of actions performed by different actors in a video sequence is presented. This work is different in two aspects with classical techniques of computer vision. On the one hand, the data relative to detected objects in the video sequence is modelled as a set of linguistic elements. On the other hand, the analysis technique is based on a Mealy machine used to represent predefined actions (behaviors to be detected) and this automaton does not obtain a relation (*object*, *action*) as result: "the object performs action ith" but instead: "the membership value of the object to action ith is Z", where Z is the final output of the sequential machine.

This paper is organized as follows. Section II describes a general classification of video analysis techniques. Later, in Section III, the transformation of data into fuzzy domain is justified and the definitions of linguistic elements used to represent the data obtained from segmentation and tracking are given. In Section IV the main idea of this new formulation of the Mealy machine, its formal definition and its transition-output table are shown. The comparison process is described in detail in Section V. Concretely, the algorithm that generates the set of input symbols for each Mealy machine. Section VI presents the obtained results in the different experiments. Finally, conclusions are given in Section VII.

II. Related Work

According to [9], the techniques to represent and recognize temporal scenarios for automatic video interpretation could be classified in different categories:

- Probabilistic and stochastic: Bayesian networks and Hidden Markov Models. The main characteristic of these techniques is to model explicitly uncertainty using numbers.
- Symbolic: action classification, automata, constraint satisfaction problem. These techniques aim at transforming numerical observations into symbolic scenarios.
- Symbolic temporal techniques: temporal constraint satisfaction problem, plan recognition, event calculus and Petri nets, chronicle recognition and temporal constraint propagation. These techniques try to model temporal relations at a symbolical level.

In activity recognition the main problems to be resolved are knowledge representation about objects, scenarios, etc. and the reasoning process. The motivation of this paper is to develop a technique that helps to interpret video sequences using as knowledge representation the fuzzy logic and approximate reasoning techniques supported by a finite state automaton. There are several works in the literature that are related to this study. For example, Hongent et al. [2] considered an activity is composed of action threads. Each single-thread action is executed by an single actor and is represented by a stochastic finite automaton of event states. Each state represents characteristics of the trajectory and shape of moving blobs. Bobick et al. [1] presents an article inspired by work in speech recognition where the inference problem is divided in two levels. The lower one obtains candidate detections of low level features and the higher one uses this values to provide an input stream for a stochastic context-free grammar parsing mechanism. Grammar and parser allows the inclusion of a priori knowledge about the structure of temporal events in a given domain.

III. Fuzzy representations of the data

In this section, a fuzzification [10] of the results obtained from tracking of moving objects in a video stream is proposed. Concretely, the information relative to the detected objects could be: the vertical and horizontal velocity of their displacement and the vertical and horizontal position in the

L. Rodriguez-Benitez is with the Escuela Politecnica, Universidad de Castilla-la Mancha, Almaden (Ciudad Real, Spain), email: luis.rodriguez@uclm.es

J. Moreno-Garcia is with the Escuela Ingeniera Tecnica Industrial, Universidad de Castilla-la Mancha, Toledo, email: juan.moreno@uclm.es

J.J. Castro, C. Solana and, L. Jimenez are with the Escuela Superior de Informatica, Universidad de Castilla-la Mancha, Ciudad Real, email: {josejesus.castro, cayetanoj.solana, luis.jimenez}@uclm.es

scene. So in this case, four linguistic variables [11] are needed to transform these concepts into a fuzzy representation. This domain change can be justified on the basis of the next arguments:

- The proposed method needs to unify into a common representation the results obtained from different tracking algorithms.
- It allows to incorporate knowledge about the analyzed scene and the motion characteristics of the candidate objects. For example, the design of the linguistic variables shown in Fig. 1 allows to differentiate between two exits doors.
- The transformation of quantitative data into qualitative values (linguistic representations) facilitates the interpretation of the information obtained from the tracking process. This fact should improve the design, codification and debug of high level vision tasks.
- After the fuzzification process, values that correspond to noise obtained from video data extraction, segmentation or tracking do not take membership values in the same fuzzy sets (labels) than data corresponding to objects detected in the scene. So, noise is easier to be characterized and then removed.

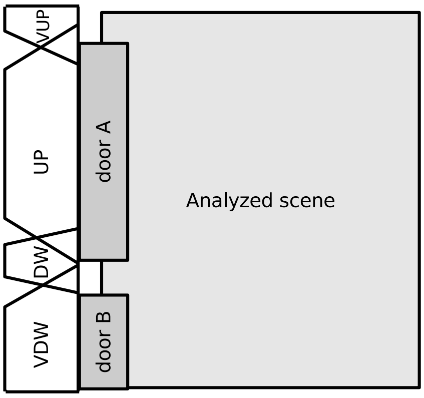

Fig. 1. Fuzzy partitions to incorporate knowledge.

A. Fuzzification of tracking information

The data obtained as result of the tracking process must be fuzzified and then a set of linguistic variables is needed. The linguistic variables used are: *vertical velocity* (VV), *horizontal velocity* (HV), *horizontal position* (HP) and *vertical position* (VP). As it was previously indicated, the design of each one of the variables depends on the scenario, characteristics of the studied objects, etc. Anyway, in this paper a set of generic linguistic variables are used as it is shown in Fig. 2 to 5.

Two fuzzy components are used to represent the information related to objects detected in the tracking process. The first one is called Linguistic blob [6] and it represents the position and the velocity of each one of the regions (from different frames) of the trajectories of the tracked objects. A **Linguistic Blob (LB)** is the 5-tuple:

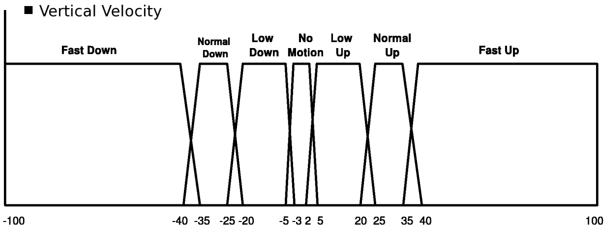

Fig. 2. Linguistic variable VV

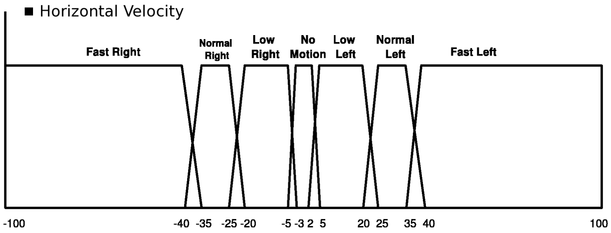

Fig. 3. Linguistic variable HV

$$< FN, I_{VH}(v_x), I_{VV}(v_y), I_{VP}(y), I_{HP}(x) > \quad (1)$$

where FN is the number of frame where the LB is detected, and the last four components ($I_{HV}(v_x)$, $I_{VV}(v_y)$, $I_{VP}(x)$, $I_{HP}(y)$) are linguistic intervals that represent the velocity and the position of the Blob. They are obtained as results of the fuzzification of the horizontal (x) and vertical (y) positions of the region and their vertical (v_y) and horizontal (v_x) velocities.

The second fuzzy representation is used to store the object trajectory and it is named as Linguistic Object [7]. A **Linguistic Object** is the tuple:

$$< IF, FF, NF, \{ListBlobs\} >$$

where IF and FF are the initial and final frames that defines the time interval during the object is present in the scene, NF is the number of frames with object motion information, and $ListBlobs$ is a list of all the Linguistic blobs that compound the object ($\{LB_{IF}, \ldots, LB_{FF}\}$). An example of a linguistic object is shown in Table I where the object moves slowly to the right (SR), its vertical position is down (D) and the object is situated at the center of the image (CH) and changes progressively to the right (R).

TABLE I
EXAMPLE OF LINGUISTIC OBJECT

Initial Frame: 2	
Final Frame = 9	
Number of Frames = 6	
LB_0 :	$\{2, \{SR : 1\}, \{NM : 1\}, \{D : 1\}, \{CH : 1\}\}$
LB_1 :	$\{3, \{SR : 1\}, \{NM : 1\}, \{D : 1\}, \{CH : 1\}\}$
LB_2 :	$\{5, \{SR : 1\}, \{NM : 1\}, \{D : 1\}, \{CH : 1\}\}$
LB_3 :	$\{6, \{SR : 1\}, \{NM, 1\}, \{D : 1\}, \{CH : 1\}\}$
LB_4 :	$\{8, \{SR : 1\}, \{NM : 1\}, \{D : 1\}, \{CH : 0.75; R : 0.25\}\}$
LB_5 :	$\{9, \{SR : 1\}, \{NM : 1\}, \{D : 1\}, \{CH : 0.75; R : 0.25\}\}$

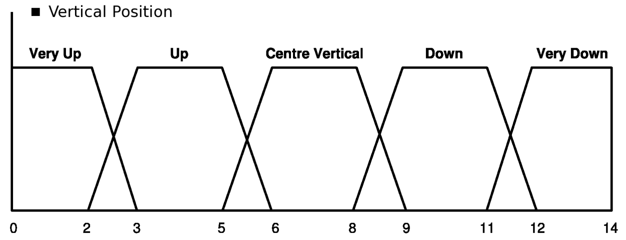

Fig. 4. Linguistic variable VP

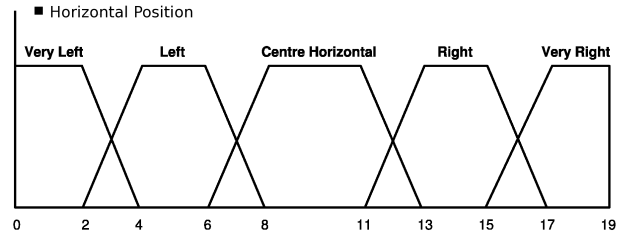

Fig. 5. Linguistic variable HP

B. *Simplification of input data*

The linguistic object is usually associated with a lot of information. To improve its interpretability and to eliminate redundant information a new fuzzy representation is proposed. It is called simplified linguistic object and it is conducted by grouping similar consecutive blobs into linguistic states. A **linguistic state** represents the position and velocity of a region in consecutive frames of a linguistic object and it is represented as:

$$LingEst =< IF,\ FF,\ SI_{HV},\ SI_{VV},\ SI_{VP},\ SI_{HP} > \quad (2)$$

where IF and FF are the initial and final frames of the state and $SI_{HV}(v_x)$, $SI_{VV}(v_y)$, $SI_{VP}(y)$, $SI_{HP}(x)$ are linguistic intervals without the membership values of the labels. Then, for example, the state $LingEst_0$ = < 11, 18, {SR; NM}, {NM}, {DW}, {VL} > could be interpretated as: "between frames 11 and 18 the object moves slowly to the right and it is situated "down" and "very left".

Given a linguistic object (*LO*), a simplified linguistic object (*SLO*) is the tuple $SLO =< SetStates >$ where *SetStates* is a set of linguistic states temporally ordered which is obtained from the atribute *ListBlobs* of a *LO*:

$$SetStates = \{LingEst_0, LingEst_1, ..., LingEst_{k-1}\} \quad (3)$$

An example is shown in Table II where it can be observed that the displacement of the object is from right to left. The reason why the time intervals associate to each state are not consecutive is caused by two factors. One is the experimental video data used in this work is obtained from compressed domain [4] and the other is the states generated by an unique blob are considered noisy states and so they are removed.

TABLE II
A SIMPLIFIED LINGUISTIC OBJECT

$LingEst_0$: <26, 26, {SR; NM}, {NM}, {DW}, {VL}>,
$LingEst_1$: <38, 41, {FR; SR}, {NM}, {DW; VDW}, {L}>,
$LingEst_2$: <59, 60, {FR; SR}, {NM}, {DW; VDW}, {R}>
$LingEst_3$: <62, 69, {FR; SR}, {NM}, {DW; VDW}, {R; VR}>
$LingEst_4$: <71, 75, {FR; SR}, {NM}, {DW; VDW}, {VR}>

IV. A MEALY MACHINE FOR RECOGNIZING ACTIONS.

The authors propose a representation of a prototype action using a finite state machine where the states of the automaton corresponds with linguistic states (Section III-B). The initial idea was to obtain a string of symbols from the information of an object detected in a video sequence. This string is employed as the input of the automaton and if it is accepted by it (finishes in a final state) then it could be considered that the automaton represents the object's behaviour.

Nevertheless, there are several factors that make the recognition process described above inviable. For example, there is a lot of noise in the tracking data: incomplete information about objects caused by occlusions, over-segmentation of objects, merge of regions, etc. Besides, there are different types of objects with very different characteristics (Fig. 6) and there are several possible trajectories associated with one only action. For example, the position of the vehicle shown in Fig. 7.b is centered around the image centre. Nevertheless, car shown in Fig. 7.a is to the left of the image.

Fig. 6. Different kinds of vehicles.

Fig. 7. Multiple trajectories in the same action.

Then the use of a Mealy machine is proposed. A Mealy machine [5] is a finite state machine that generates an output based on its current state and an input. The last output generated by the automaton is a membership value of the object to the action represented by the sequential machine. This value is taken into account independently of the automaton finishes in a final state or not and it takes values between 0 and 1.

A Mealy Machine used to obtain a membership value of an Object to an Action (*MMOA*) is a 6-tuple,

$$MMOA = <\Sigma_E,\ \Sigma_S,\ Q,\ f,\ g,\ q_0)> \tag{4}$$

consisting of the following:

- A finite set called the input alphabet Σ_E.
  Each input symbol is a set consisting of three elements (d_a, d_{a+1}, s') where d_a is the distance value between the active state of the automaton and a linguistic state of the object, d_{a+1} is the distance value between the next state of the finite states machine and the same linguistic state, and *s'* is the last output value of the sequential machine.
- A finite set called the output alphabet Σ_S.
- A finite set of states Q: $\{q_0,\ q_1,\ ...\ ,\ q_n,\ q_{n+1}\}$ where $q_1 = LingEst_0$, $q_2 = LingEst_1$, ..., and $q_n = LingEst_{n-1}$ with $\{LingEst_0,\ LingEst_1,$ ..., $LingEst_{n-1}\} \in PrototypeAction$, q_0 is the initial state and, q_{n+1} is a draining state.
- A transition function $f : Q\ X\ \Sigma_E \longrightarrow Q$.
- An output function $g : Q\ X\ \Sigma_E \longrightarrow \Sigma_S$.
- A start state (also called initial state) q_0 which is an element of *Q*.

The transition-output table of the *MMOA* is shown in Tables III to IV and it shows the desired next-state variable combination for each state/input combination. For example, if in the position (q_a, $Condition_y$) is q_{a+1}/s, the active state is q_a, the input symbol satisfies $Condition_y$ and then the sequential machine changes to state q_{a+1}. This transition produces the output symbol s.

Now, the conditional expressions of the transition function of the automaton will be studied in more detail. They are identified by alphabetic characters (*a*, *b* and, *c*) as it is shown in Fig. 8 where it can be observed a sequential machine that satisfies the definition of MMOA (Equation 4). First, symbols not previously defined are explained below:

- TD is the total distance. This distance [7] is a Euclidean distance defined between the position of labels in the linguistic variable.
- $LingEst_c$ is a linguistic state of the studied object.
- $d_a = TD(q_a, LingEst_c)$. Total distance between the active state and $LingEst_c$.
- $d_{aPREV} = TD(q_a, LingEst_{c-1})$. Total distance between the active state and a previous linguistic state of the object ($LingEst_{c-1}$).
- $d_{a+1} = TD(q_{a+1}, LingEst_c)$. Total distance between the next state and $LingEst_c$.
- *ThresholdD* is a configuration variable that takes values between 0 and 1.

Now, the transitions of the automaton will be described:

- **a** $\equiv d_a < d_{a+1}$ AND $d_a \leq ThresholdD$
  The sequential machine remains in the state q_a if $LingEst_c$ is more similar to q_a than to q_{a+1}. Then the output symbol generated s results of evaluating the expression $s = s' - d_{aPREV} + min(d_{aPREV}, d_a)$ where *s'* is a variable that stores previous output of the automaton. Using this expression it is possible to take into account only the minimum output when the finite states machine transits to the same state.
- **b** $\equiv d_{a+1} \leq d_a$ AND $d_{a+1} \leq ThresholdD$
  There is a transition from state q_a to state q_{a+1} if $LingEst_c$ is more similar to q_{a+1} than to q_a. The output symbol generated s results on evaluating the expression $s = s' + d_{a+1}$, that is, previous output (*s'*) plus the distance to state q_{a+1}.
- **c** $\equiv d_a > ThresholdD$ AND $d_{a+1} > ThresholdD$
  There is a transition to the draining state when the object motion finishes or when $LingEst_c$ is neither similar to d_a nor d_{a+1}.

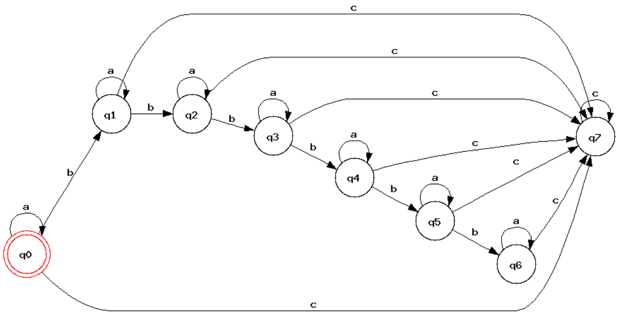

Fig. 8. Mealy machine representing an action.

TABLE III
TRANSITION-OUTPUT TABLE (TRANSITION 'A').

q_a	**a** ($d_a < d_{a+1}$ AND $d_a \leq ThresholD$)
q_0	$q_0/s = 0$
q_1	$q_1/s = s' - d_{aPrev} + min(d_{aPrev}, d_1)$
q_2	$q_2/s = s' - d_{aPrev} + min(d_{aPrev}, d_2)$
.	.
.	.
q_{n-1}	$q_{n-1}/s = s' - d_{aPrev} + min(d_{aPrev}, d_{n-1})$
q_n	$q_n/s = s' - d_{aPrev} + min(d_{aPrev}, d_n)$
q_{n+1}	ϕ

TABLE IV
TRANSITION-OUTPUT TABLE (TRANSITIONS 'B' AND 'C')

q_a	**b** ($d_{a+1} \leq d_a$ AND $d_{a+1} \leq ThresholD$)	**c** ($d_a > ThresholD$ AND $d_{a+1} > ThresholD$)
q_0	$q_1/s = s' + d_1$	$q_{n+1}/s = s' + d_0$
q_1	$q_2/s = s' + d_2$	$q_{n+1}/s = s' + d_1$
q_2	$q_3/s = s' + d_3$	$q_{n+1}/s = s' + d_2$
.	.	.
.	.	.
q_{n-1}	$q_n/s = s' + d_n$	$q_{n+1}/s = s' + d_{n-1}$
q_n	ϕ	$q_{n+1}/s = s' + d_n$
q_{n+1}	ϕ	$q_{n+1}/s = s' + d_{n+1}$

V. COMPARING OBJECT - PROTOTYPE ACTIONS

In this section the process of comparing the behaviour of an object with every prototype action is described. The comparison results are stored in a set called *membership list*

(*ML*). Its ith element (ML_i) corresponds to the membership of the object to the action *i*.

Fig. 9 shows the comparison process. It takes as input an object represented as a simplified linguistic object and the temporal data is extracted from the linguistic states and represented as a set of events. The initial and final event represent the appearance and disappearance of the object in the video sequence respectively. The linguistic states of the object and the set of events are used as input of Algorithm 1. Besides, the information about the active state and the last output of a MMOA is needed. This algorithm generates a set of input symbols for each one of the Mealy machines. The final event establishes the end of the comparison process and the last output symbol of each sequential machine is a distance value (*D*) transformed into a similarity value stored in *ML*.

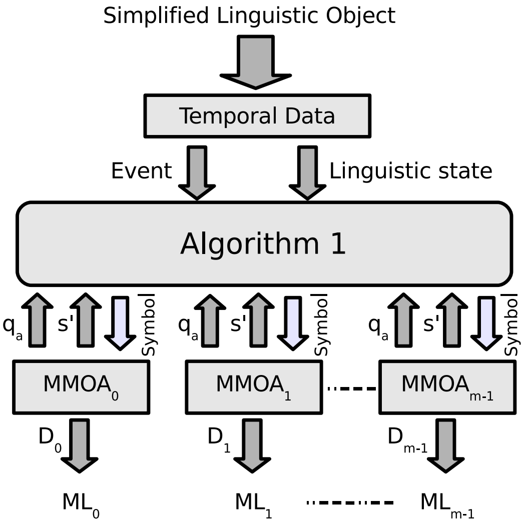

Fig. 9. General schema of comparison process

Algorithm 1 Generation of MMOA input symbols.

```
INPUT: LingEst (a linguistic state of the object).
INPUT: $EV$ (an event).
INPUT: $q_a$ (active state of MMOA)
INPUT: s' (last output of MMOA)
OUTPUT: InputSymbol=($d_a$; $d_{a+1}$; s)

if (EV!=FinalEvent) then
  {if the event is not final}
  if ($q_a$=$q_0$) then
    InputSymbol $\leftarrow (ThresholdD;\ TD(LingState, q_1);\ 0)$
    {looking for the first transition}
  end if
  if ($q_a$=$q_{n+1}$) then
    InputSymbol $\leftarrow (1;\ 1;\ s')$ {continues in state $q_{n+1}$}
  end if
  if ($q_a$=$q_n$) then
    InputSymbol $\leftarrow (TD(LingEst, q_a);\ 1;\ s')$ {continues
    in state $q_n$ or changes to state $q_{n+1}$}
  end if
  if ($q_a \neq q_0$ AND $q_a \neq q_n$ AND $q_a \neq q_{n+1}$) then
    InputSymbol$\leftarrow (TD(LingEst, q_a); TD(LingEst, q_{a+1});$
    $s')$ {compares active and active+1 state}
  end if
else
  {if event is final}
  if (($q_a \neq q_n$) AND ($q_a \neq q_{n+1}$)) then
    InputSymbol $\leftarrow (n-a;\ n-a;\ s')$ {mandatory transi-
    tion to $q_{n+1}$}
  end if
  if ($q_a = q_{n+1}$) then
    InputSymbol $\leftarrow (1;\ 1;\ s')$ {continues in state $q_{n+1}$}
  end if
end if
```

In Table VI an example of the comparison between a prototype action and an object (Table V) is shown. The configuration variable *Threshold* is equal to 0.5.

The comparison process obtains a numerical value representing a distance between the object and the action. Nevertheless, the obtained value must be a similarity measure from range 0 to 1. This can be achieved using Equation 5, where D is the final output of the automaton and $|MMOA|$ is the number of states of MMOA. This equation takes into account that states q_0 and q_{n+1} does not represents linguistic states of the prototype action and they are not compared with the object.

$$ML \leftarrow 1 - \frac{D}{|MMOA| - 2} \tag{5}$$

TABLE V
PROTOTYPE ACTION AND OBJECT COMPARED USING A MMOA

ACTION

q_1 =<{SR}, {NM}, {DW; VDW}, {VL ;L}>
q_2 =<{SR}, {NM}, {DW; VDW}, {L}>
q_3 =<{SR}, {NM}, {DW; VDW}, {L; CH}>
q_4 =<{SR}, {NM}, {DW; VDW}, {CH; R}>
q_5 =<{SR}, {NM}, {DW; VDW}, {R; VR}>
q_6 =<{SR}, {NM}, {DW; VDW}, {VR}> }

OBJECT

$LingEst_0$: <26, 26, {SR; NM}, {NM}, {DW}, {VL}>,
$LingEst_1$: <38, 41, {FR; SR}, {NM}, {DW; VDW}, {L}>,
$LingEst_2$: <59, 60, {FR; SR}, {NM}, {DW; VDW}, {R}>
$LingEst_3$: <62, 69, {FR; SR}, {NM}, {DW; VDW}, {R; VR}>
$LingEst_4$: <71, 75, {FR; SR}, {NM}, {DW; VDW}, {VR}>

TABLE VI
A COMPARISON EXAMPLE

Event	Active State	Object State	Symbol	Transit./Output
{26, I}	q_0	$EstLing_0$	(0.5; 0, 0)	q_1/0
{38, M}	q_1	$EstLing_1$	(0; 0; 0)	q_2/0
{59, M}	q_2	$EstLing_2$	(1; 0.5; 0)	q_3/0.5
{62, M}	q_3	$EstLing_3$	(0; 0; 0.5)	q_4/0.5
{71, M}	q_4	$EstLing_4$	(0; 0; 0.5)	q_5/0.5
{75, F}	q_5	ϕ	(1; 1; 0.5)	q_7/1.5

VI. Experimental results

The set of experiments (Table VII) try to identify the behavior of vehicles in traffic scenes. First one (Fig. 10.a) is a vehicle crossing and second one (Fig. 10.b) is a three lane highway, where central lane allows turns to be made in both directions. The major road flow exceeds 2200 vehicles/hour. A web cam located inside a car to record the traffic scenes is used. The position of the camera is different if it is compared to applications of traffic monitoring (they usually work with aerial scenes). The sampling frequency is 30 frames per second with resolution 320X240 pixels.

TABLE VII
Description of the experiments.

Experiment	**Duration** (minutes)	**Size** (megabytes)	**Frames**
1	15.14	65.41	27250
2	4.17	18.1	7730

Fig. 10. Pictures of experiments.

Seven prototype actions are used in the first experiment and five in the second one. So, seven and five membership values will be obtained respectively from the comparison between an object and all the actions represented by means of an automaton. The action being performed by the object is the one that maximizes the membership value (*ML* obtained from each MMOA). Furthermore, the membership value must be greater than a configuration variable called *MinMembership*.

To evaluate performance of action recognition process the situations to be considered are:

1) True-Positive (TP): the solution provided by the system leads to a right action.
2) False-Positive (FP): the system's solution is not an action taken by the object.
3) True-Negative (TN): when an erroneous output of the tracking algorithm is not associated with any of the predefined actions.

True-Positives and True-Negatives are good results for the system. On the contrary, False-Positives are wrong results. System evaluation measures are defined using Equations 6 and 7 [3].

$$Detection\ Probability \leftarrow \frac{TP+TN}{Total\ Objects} \tag{6}$$

$$Precision \leftarrow \frac{TP+TN}{TP+TN+FP} \tag{7}$$

Before the detailed analysis of results, it must be indicated that input data of the video analysis technique is obtained from the output of a tracking algorithm [8]. The results of the tracking phase are shown in Table VIII.

TABLE VIII
Evaluation of tracking results used as input data.

Exp.	**Objects**	**TP**	**FP**	**D. Probability**	**Precision**
1	141	141	82	100%	63.23%
2	120	117	16	97.78%	88%

The aim of the video analysis should be the determination of the behavior of correctly detected objects (true-positives in tracking) and the no establishment of relationship between actions and wrong detections (false-positives in tracking). Table IX shows the results of both experiments. The better results are obtained with *MinMembership* equal to 0.4 and the main difference between the two experiments is that results of tracking are worse for experiment 2. Nevertheless, 54 true-negatives are detected by the comparison process

TABLE IX
Evaluation of video action recognition.

Exp.	**Objects**	**TP**	**FP**	**TN**	**D. Probability**	**Precision**
1	141	71	38	54	88.65%	76.69%
2	120	102	16	3	87.5%	86.78%

VII. Conclusions

From theory and experiments presented in this paper it can be concluded that: Fuzzy logic successfully manages the inherent uncertainty of the results obtained by the tracking algorithms. The Mealy machine is able to identify wrong detections of medium-level vision tasks (segmentation and tracking). The action recognition technique obtains good results even when the prototype actions are selected directly without using any training data and learning algorithms. Predefined actions are represented by means of linguistic representations. This fact allows final users to define manually the set of prototype actions. The membership value of an object to an action gives extra information about the similarity between the pair (object, action). Although in this paper tracking data used is represented by means of four linguistic variables (*HV*, *VV*, *VP* and, *HP*), the system designed is highly scalable with respect of the type and the amount of variables used.

Acknowledgement

This work has been funded by the Spanish Ministry of Education and Science (TIN2007-62568) and the Regional Government of Castilla-la Mancha (PAC06-0141 and PBC06-0064).

References

[1] A. F. Bobick and Y. Ivanov. "Action Recognition Using Probabilistic Parsing", Proc. of CVPR'98, Santa Barbara, California, pp. 196-202. 1998.

[2] S. Hongeng , R. Nevatia and, F. Bremond, Video-based event recognition: activity representation and probabilistic recognition methods, *Computer Vision and Image Understanding*, v.96 n.2, p.129-162, November 2004

[3] Perera, A. G. Amitha; Hoogs, Anthony; Srinivas, Chukka; Brooksby, Glen; Hu, Wensheng. Evaluation of Algorithms for Tracking Multiple Objects in Video. *Applied Imagery and Pattern Recognition Workshop*, 2006. AIPR 2006.

[4] Moving Picture Experts Group. The MPEG home page. Available in: *http://www.chiariglione.org/mpeg/*.

[5] GH. Mealy. A Method to Synthesizing Sequential Circuits. Bell System Technical J, 1045-1079. (1955).

[6] L. Rodriguez-Benitez, J. Moreno-Garcia, J.J. Castro-Schez and, L. Jimenez, Linguistic Motion Description for an Object on MPEG Compressed Domain, In *Proceedings of Eleventh International Fuzzy Systems Association World Congress*, International Fuzzy Systems Association, 2005.

[7] L. Rodriguez-Benitez, J. Moreno-Garcia, J.J. Castro-Schez, and L. Jimenez, An approximate reasoning technique for segmentation on compressed MPEG video, In *Proceedings of VISAPP, 2nd International Conference on Computer Vision Theory and Applications*, 2007.

[8] L. Rodriguez-Benitez, J. Moreno-Garcia, J.J. Castro-Schez, and L. Jimenez, Fuzzy logic to track objects from MPEG video sequences, To be published In *Proceedings of IPMU, Information processing and management of uncertainty in Knowledge-based systems*, 2008. (Available in http://oreto.inf-cr.uclm.es).

[9] V. Van-thinh, Temporal scenario for automatic video interpretation, Doctoral Thesis, 2004.

[10] L.A. Zadeh, *Fuzzy Set*, Information and Control, 1960.

[11] L.A. Zadeh, The concept of a linguistic variable and its applications to approximate reasoning, *Information Science*, 1975.

Luis Rodriguez-Benitez graduated with an MS degree from the Computer Science Department at the University of Granada (Spain) in 1997. He received PhD degree from the University of Castilla-La Mancha in 2008. He is an Assistant Professor of Computer Science at the University of Castilla-La Mancha, Almaden (Spain). The interest of his current research includes computer vision, decision support, movements recognition and fuzzy and linguistic modeling.

Juan Moreno-Garcia is an Associate Professor of Industrial Engineering at the University of Castilla-La Mancha, Toledo (Spain). He received a BE degree from the University of Castilla-La Mancha in 1992, M.S. degree from the University of Murcia in 1996 and Ph.D degree from the University of Castilla-La Mancha in 2002. His main fields of interest are fuzzy and linguistic modelling, dynamic systems modelling, fuzzy logic, neural networks. Currently he is working in soft-computing applied to cognitive

Jose J. Castro-Schez received the M.S. degree in 1995 and Ph.D. degree in 2001 from the Computer Science Department at the University of Granada (Spain). He is an Associate Professor of Computer Science at the University of Castilla-La Mancha, Ciudad Real (Spain). His research interests include: knowledge acquisition, machine learning, decision support, electronic commerce and issues of representation in AI. He is author of numerous papers on AI-related subjects.

Cayetano Solana received the M.S. degree in 2007 from the Computer Science Department at the University of Castilla-la Mancha (Spain). He is a doctoral student of Computer Science at the University of Castilla-La Mancha, Ciudad Real (Spain). His research interests include: electronic democracy, decision support and multi-agent decision systems.

Luis Jimenez is an Associate Professor of Computer Science at the University of Castilla-La Mancha, Ciudad Real (Spain), where he founded and directs the ORETO Group. He received the MS degree in 1991 and PhD degree in 1997 from the Computer Science Department at the University of Granada (Spain). His main fields of interest are fuzzy logic, knowledge-based systems, machine learning and related applications, leading several research projects on these topics. He is a member of European Society of Fuzzy Logic and Technology (EUSFLAT).