Incremental Video Event Learning

Marcos Zúñiga[1,*], François Brémond[2], and Monique Thonnat[2]

[1] Electronics Department - UTFSM, Av. España 1680, Valparaíso, Chile
marcos.zuniga@usm.cl
http://profesores.elo.utfsm.cl/~mzuniga

[2] INRIA - Projet PULSAR, 2004 rte. des Lucioles, B.P. 93, 06902 Sophia Antipolis Cedex, France
Francois.Bremond@sophia.inria.fr, Monique.Thonnat@sophia.inria.fr
http://www-sop.inria.fr/pulsar

Abstract. We propose a new approach for video event learning. The only hypothesis is the availability of tracked object attributes. The approach incrementally aggregates the attributes and reliability information of tracked objects to learn a hierarchy of state and event concepts. Simultaneously, the approach recognises the states and events of the tracked objects. This approach proposes an automatic bridge between the low-level image data and higher level conceptual information. The approach has been evaluated for more than two hours of an elderly care application. The results show the capability of the approach to learn and recognise meaningful events occurring in the scene. Also, the results show the potential of the approach for giving a description of the activities of a person (e.g. approaching to a table, crouching), and to detect abnormal events based on the frequency of occurrence.

1 Introduction

Video event analysis has become one of the biggest focus of interest in the video understanding community [3]. The interest of researchers has been mainly focused on the recognition of predefined events [8] and off-line learning of the events [9]. To date, very little attention has been given to incremental event learning in video [5], which should be the natural step further real-time applications for handling unexpected events.

We propose **MILES** (**M**ethod for **I**ncremental **L**earning of **E**vents and **S**tates), a new event learning approach, which aggregates on-line the **attributes** and **reliability information** of tracked objects (e.g. people) to **learn** a hierarchy of concepts corresponding to **events**. Reliability measures are used to focus the learning process on the most valuable information. Simultaneously, MILES **recognises** new occurrences of events previously learned. The only hypothesis of MILES is the availability of tracked object attributes, which are the needed

[*] This research has been supported in part by the Science and Technology Research Council of Chile (CONICYT) in the framework of INRIA (Sophia Antipolis) and CONICYT cooperation agreement.

M. Fritz, B. Schiele, and J.H. Piater (Eds.): ICVS 2009, LNCS 5815, pp. 403–414, 2009.

input for the approach. This work centres its interest in learning events in general, and is validated for specific events for home-care (e.g. approaching to a table, crouching).

MILES is an incremental approach, as no extensive reprocessing is needed upon the arrival of new information. The incremental aspect is important as the available training examples can be insufficient for describing all the possible scenarios in a video scene. Also, incremental approaches are suitable for on-line learning, as the processing cost is very low.

This approach proposes an automatic bridge between the low-level image data and higher level conceptual information. The learnt events can serve as building blocks for higher level behavioural analysis. The main novelties of the approach are the capability of learning events in general, the utilisation of a explicit quality measure for the built event hierarchy, and the consideration of measures to focus learning in reliable data.

This paper is organised as follows. After the state-of-the-art, Section 3 describes the proposed event learning approach and illustrates the learning procedure. Then, Section 4 presents the results obtained for an elderly care application.

2 State-of-the-Art

Most of video event learning approaches are supervised using general techniques as Hidden Markov Models (HMM) and Dynamic Bayesian Network (DBN) [2], requesting annotated videos representative of the events to be learned. Few approaches can learn events in an unsupervised way using clustering techniques. For example, in [9] the authors propose a method for unusual event detection, which first clusters a set of seven blob features using a Gaussian Mixture Model, and then represents behaviours as an HMM, using the cluster set as the states of the HMM. In [6], the authors propose an approach for learning events using spatial relationships between objects (e.g. the proximity between a person and kitchen dishes) in an unsupervised way, but performed off-line. Also, in [7], they learn in an unsupervised way composite events using the APRIORI clustering algorithm. This technique requires the manual definition of the simple events to build the composite ones. However, these unsupervised clustering techniques request to (re)process off-line (not real-time) the whole cluster distribution.

Some other techniques can learn on-line the event model by taking advantage of specific event distributions. For example, in [5], the authors propose a method for incremental trajectory clustering by mapping the trajectories into the ground plane decomposed in a zone partition. A new trajectory cluster is created as soon as the trajectory extremity does not belong to any of the zone corresponding to a trajectory cluster. Their approach performs learning only on spatial information, it cannot take into account time information, and do not handle noisy data.

Therefore, a new approach for incremental event learning is needed to handle unexpected situations.

3 MILES: A New Approach for Incremental Event Learning and Recognition

MILES is based on *incremental concept formation models* [1]. Conceptual clustering consists in describing classes by first generating their conceptual descriptions and then classifying the entities according to these descriptions. *Incremental concept formation models* is a conceptual clustering approach which incrementally creates a new concept without extensive reprocessing of the previously encountered instances. The knowledge is represented by a hierarchy of concepts partially ordered by generality. A *category utility* function is used to evaluate the quality of the obtained concept hierarchies [4].

MILES is an extension of incremental concept formation models for learning video events. The approach uses as input a set of attributes from the tracked objects in the scene. Hence, the only hypothesis of MILES is the availability of tracked object attributes (e.g. position, posture, class, speed). MILES constructs a **hierarchy of state and event concepts h**, based on the **state and event instances** extracted from the tracked object attributes.

A **state concept** is the modelisation of a spatio-temporal property valid at a given instant or stable on a time interval. A **state concept** $S^{(c)}$, in a hierarchy $\mathbf{h}$, is modelled as a **set of attribute models** $\{n_i\}$, with $i \in \{1,..,T\}$, where n_i is modelled as a random variable N_i which follows a Gaussian distribution $N_i \sim \mathcal{N}(\mu_{n_i}; \sigma_{n_i})$. T is the number of attributes of interest. The state concept $S^{(c)}$ is also described by its **number of occurrences** $N(S^{(c)})$, its **probability of occurrence** $\mathcal{P}(S^{(c)}) = N(S^{(c)})/N(S^{(p)})$ ($S^{(p)}$ is the root state concept of $\mathbf{h}$), and the **number of event occurrences** $N_E(S^{(c)})$ (number of times that state $S^{(c)}$ passed to another state, generating an event).

A **state instance** is an instantiation of a state concept, associated to a tracked object $\mathbf{o}$. The state instance $S^{(o)}$ is represented as the set attribute-value-measure triplets $\mathbf{T_o} = \{(v_i; V_i; R_i)\}$, with $i \in \{1, \ldots, T\}$, where R_i is the reliability measure associated to the obtained value V_i for the attribute v_i. The measure $R_i \in [0, 1]$ is 1 if associated data is totally reliable, and 0 if totally unreliable.

An **event concept** $E^{(c)}$ is defined as the change from a starting state concept $S_a^{(c)}$ to the arriving state concept $S_b^{(c)}$ in a hierarchy $\mathbf{h}$. An **event concept** $E^{(c)}$ is described by its **number of occurrences** $N(E^{(c)})$, and its **probability of occurrence** $\mathcal{P}(E^{(c)}) = N(E^{(c)})/N_E(S_a^{(c)})$ (with $S_a^{(c)}$ its starting state concept).

The state concepts are hierarchically organised by generality, with the children of each state representing specifications of their parent. A unidirectional link between two state concepts corresponds to an event concept. An example of a hierarchy of states and events is presented in Figure 1. In the example, the state S_1 is a more general state concept than states $S_{1.1}$ and $S_{1.2}$, and so on. Each pair of state concepts ($S_{1.1}$; $S_{1.2}$) and ($S_{3.2}$; $S_{3.3}$), is linked by two events concepts, representing the occurrence of events in both directions.

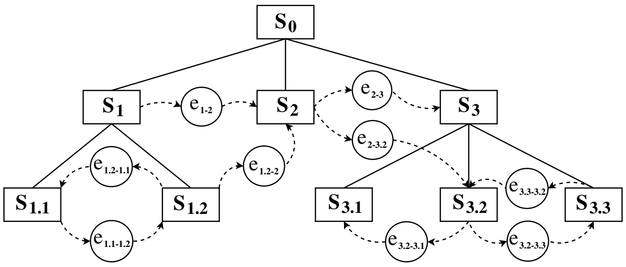


Fig. 1. Example of a hierarchical event structure resulting from the proposed event learning approach. Rectangles represent states, while circles represent events.

3.1 MILES Learning Process

The input of MILES corresponds to a list of tracked object attributes. MILES needs that the objects are tracked in order to detect the occurrence of *events.* There is no constraint on the number of attributes, as MILES has been conceived for learning state and event concepts in general. For each attribute, MILES needs a normalisation value to be defined prior to its computation. This value corresponds to the concept of *acuity.*

The **acuity** [1] is a system parameter that specifies the minimal value for numerical attributes standard deviation σ in a state concept. In psychophysics, the *acuity* corresponds to the notion of a *just noticeable difference*, the lower limit on the human perception ability. This concept is used for the same purpose in MILES, but the main difference with its utilisation in previous work [1] is that the *acuity* was used as a single parameter, while in MILES each numerical attribute n_i has associated an acuity value A_{n_i}. This improvement allows to represent different normalisation scales and units associated to different attributes (e.g. kilo, meter, centimetre) and to represent the interest of users for different applications (more or less coarse precision). The acuity parameter needs to be set-up manually to enable the user to regulate the granularity of the learned states.

Initially, before the first execution of MILES, the hierarchy **h** is initialised as an empty tree. If MILES has been previously executed, the incremental nature of MILES learning process allows that the resulting hierarchy **h** can be utilised as the initial hierarchy of a new execution.

At each video frame, MILES utilises the list of all tracked objects **O** for updating the hierarchy **h**. For each object **o** in **O**, MILES first gets the set of triplets $\mathbf{T_o}$, which serves as input for the state concept updating process of **h**. This updating process is described in Section 3.2. The updating process returns a list $\mathbf{L_o}$ of the current state concepts recognised for the object **o** at each level of **h**.

Then, the event concepts $E^{(c)}$ of the hierarchy **h** are updated comparing the new state concept list L_o with the list of state concepts recognised for the object **o** at the previous frame.

Finally, MILES gives as output for each video frame, the updated hierarchy **h** and the list of the currently recognised state and event concepts for each object **o** in **O**.

3.2 States Updating Algorithm

The state concept updating algorithm is described below by pseudo-code.

```
list_of_states updateStates (hierarchy h, list_of_triplets To) {
      list_of_states L;
      C = getRootOfHierarchy( h );
      if emptyTree( h ) {
         insertRoot( h, To );
      } else if isTerminalState( C ) {
         if cutoffTestPassed ( C, To )
            createNewTerminals( h, To );
         incorporateState( C, To );
      } else {
         incorporateState( C, To );
         P = highestScoreState( h );
         Q = newStateConcept( h, To );
         R = secondScoreState( h );
         y = mergeCategoryUtilityScore( P, R );
         z = splitCategoryUtilityScore( P});
         if score( P ) is bestScore
            updateStates( getSubTree( h, P ), To );
         else if score( Q ) is bestScore
            insertChild (Q, h);
         else if y is bestScore {
            M = mergeStates( P, R, h );
            updateStates(getSubTree( h, M), To );
         } else if z is bestScore {
            splitStates( P, h );
            updateStates( getSubTree( h, C), To );
         }
      }
      insertCurrentState( C, L );
      return L;
}
```

The algorithm starts by accessing the analysed state **C** from the current hierarchy **h** (with function *getRootOfHierarchy*, which returns the root state of **h**). The initialisation of the hierarchy is performed by creating a state with the triplets $\mathbf{T_o}$, for the first processed object.

Then, for the case that **C** corresponds to a terminal state (the state has no children), a *cutoff* test is performed (function *cutoffTestPassed*).

The **cutoff** is a criteria utilised for stopping the creation (i.e. specialisation) of children states. It can be defined as:

$$\text{cutoff} = \begin{cases} \text{true} & \text{if} \quad \{ \quad \mu_{n_i} - V_{n_i} \leq A_{n_i} \\ & \qquad\qquad | \quad \forall\ i \in \{1,..,T\} \quad \} \ , \\ \text{false} & \text{else} \end{cases} \tag{1}$$

where V_{n_i} is the value of the i-th triplet of $\mathbf{T_o}$. This equation means that the learning process will stop at the concept state $S_k^{(c)}$ if no meaningful difference exists between each attribute value of $\mathbf{T_o}$ and the mean value μ_{n_i} of the attribute n_i for the state concept $S_k^{(c)}$ (based on the attribute *acuity* A_{n_i}).

If the *cutoff* test is passed, the function *createNewTerminals* generates two children for $\mathbf{C}$, one initialised with $\mathbf{T_o}$ and the other as a copy of $\mathbf{C}$. Then, passing or not passing the *cutoff* test, $\mathbf{T_o}$ is incorporated to the state $\mathbf{C}$ (function *incorporateState* described in Section 3.3). In this terminal state case, the updating process then stops.

If $\mathbf{C}$ has children, first $\mathbf{T_o}$ is immediately incorporated to $\mathbf{C}$. Next, different new hierarchy configurations have to be evaluated among all the children of $\mathbf{C}$. In order to determine in which state concept the triplets list $\mathbf{T_o}$ is next incorporated (i.e. the state concept is recognised), a quality measure for state concepts called **category utility** is utilised, which measures how well the instances are represented by a given category (i.e. state concept).

The category utility CU for a class partition of K state concepts (corresponding to a possible configuration of the children for the currently analysed state $\mathbf{C}$) is defined as:

$$CU = \frac{\displaystyle\sum_{k=1}^{K} \frac{\mathcal{P}(S_k^{(c)}) \displaystyle\sum_{i=1}^{T} \left(\frac{A_{n_i}}{\sigma_{n_i}^{(k)}} - \frac{A_{n_i}}{\sigma_{n_i}^{(p)}} \right)}{2\cdot T \cdot \sqrt{\pi}}}{K} , \tag{2}$$

where $\sigma_{n_i}^{(k)}$ (respectively for $\sigma_{n_i}^{(p)}$) is the standard deviation for the attribute n_i of $\mathbf{T_o}$, with $i \in \{1,2,..,T\}$, in the state concept $S_k^{(c)}$ (respectively for the root state $S_p^{(c)}$).

It is worthy to note that the category utility CU serves as the major criteria to decide how to balance the states given the learning data. CU is an efficient criteria because it compares the relative frequency of the candidate states together with the relative Gaussian distribution of their attributes, weighted by their significant precision (predefined acuity).

Then, the different alternatives for the incorporation of $\mathbf{T_o}$ are:

(a) The incorporation of $\mathbf{T_o}$ to a existing state $\mathbf{P}$ gives the best CU score. In this case, the function *updateStates* is recursively called, considering $\mathbf{P}$ as root.

(b) The generation of a new state concept $\mathbf{Q}$ from instance $\mathbf{T_o}$ gives the best CU score $\mathbf{x}$. In this case, the function *insertChild* inserts the new state $\mathbf{Q}$ as child of $\mathbf{C}$, and the updating process stops.

(c) Consider the state $\mathbf{M}$ as the resulting state from merging the best state $\mathbf{P}$ and the second best state $\mathbf{R}$. Also, consider $\mathbf{y}$ as the CU score of replacing states $\mathbf{P}$ and $\mathbf{R}$ with $\mathbf{M}$. If the best CU score is $\mathbf{y}$, the hierarchy is modified by the **merge operator** (function *mergeStates*). Then, *updateStates* is recursively

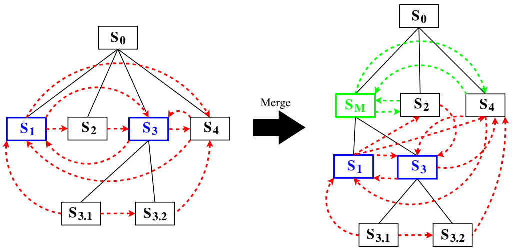


Fig. 2. Result of a merging operation. Blue boxes represent the states to be merged. The green box represents the resulting merged state. Red dashed lines represent the existing events, while the green dashed lines are the new events from the merging process.

called, using the subtree from state **M** as the tree to be analysed. The **merge operator** consists in merging two state concepts S_p and S_q into one state S_M, while S_p and S_q become the children of S_M, and the parent of S_p and S_q becomes the parent of S_M, as depicted in Figure 2. The merge operator also generates new events for state S_M which generalise the transitions incoming and leaving states S_p and S_q. (d) Consider **z** as the CU score of replacing state **P** with its children. If the best CU score is **z**, the hierarchy is modified by the **split operator** (function *splitStates*). Then, *updateStates* is called, using the subtree from the current state **C** again. The **split operator** consists in replacing a state S with its children, as depicted in Figure 3. This process implies to suppress the state concept S together with all the events in which the state is involved. Then, the children of the state S must be included as children of the parent state of S.

At the end of function *updateStates*, each current state **C** for the different levels of the hierarchy is stored in the list **L** of current state concepts for object **o**, by the function *insertCurrentState*.

3.3 Incorporation of New Object Attribute Values

The incorporation process consists in updating a state concept with the triplets $\mathbf{T_o}$ for an object **o**. The proposed updating functions are incremental in order to improve the processing time performance of the approach. The incremental updating function for the mean value μ_n of an attribute n is presented in Equation (3).

$$\mu_n(t) = \frac{V_n \cdot R_n + \mu_n(t-1) \cdot Sum_n(t-1)}{Sum_n(t)}, \tag{3}$$

with

$$Sum_n(t) = R_n + Sum_n(t-1), \tag{4}$$

where V_n is the attribute value and R_n is the reliability. Sum_n is the accumulation of reliability values R_n.

The incremental updating function for the standard deviation σ_n for attribute n is presented in Equation (5).

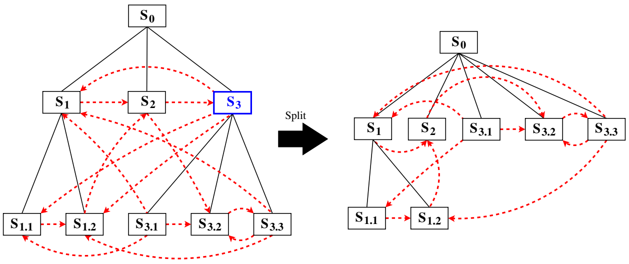


Fig. 3. Split operator in MILES approach. The blue box represents the state to be split. Red dashed lines represent events.

$$\begin{aligned} &\sigma_n(t) = \sqrt{\tfrac{Sum_n(t-1)}{Sum_n(t)} \cdot \left(\sigma_n(t-1)^2 + \tfrac{R_n \cdot \Delta_n}{Sum_n(t)}\right)}. \\ &with \\ &\Delta_n = (V_n - \mu_n(t-1))^2 \end{aligned} \tag{5}$$

For a new state concept, the initial values taken for Equations (3), (4), and (5) with $t = 0$ correspond to $\mu_n(0) = V_n$, $Sum_n(0) = R_n$, and $\sigma_n(0) = A_n$, where A_n is the *acuity* for the attribute n.

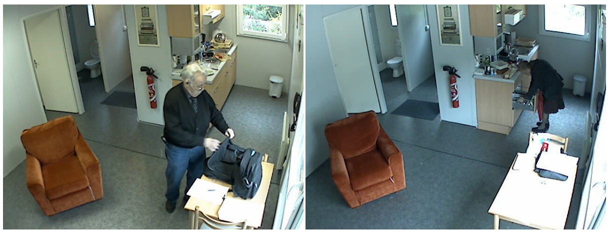
In case that, after updating the standard deviation Equation (5), the value of $\sigma_n(i)$ is lower than the *acuity* A_n, $\sigma_n(i)$ is reassigned to A_n. This way, the acuity value establishes a lower bound for the standard deviation of an attribute.

4 Evaluation of the Approach

The capability of MILES for automatically learning and recognising real world situations has been evaluated, using two videos for elderly care at home. The video scene corresponds to an apartment with a table, a sofa, and a kitchen, as shown in Figure 4. The videos correspond to an elderly man (Figure 4a) and an elderly woman (Figure 4b), both performing tasks of everyday life as cooking, resting, and having lunch. The lengths of the sequences are 40000 frames (approximately 67 minutes) and 28000 frames (approximately 46 minutes).

The input information is obtained from a tracking method which computes reliability measures to object attributes, which is not included due to space constraints.

The attributes of interest for the evaluation are 3D position (x, y), an attribute for standing or crouching posture, and interaction attributes $SymD_{table}$, $SymD_{sofa}$, and $SymD_{kitchen}$ between the person and three objects present in the scene (table, sofa, and kitchen table). For simplicity, The interaction attributes are represented with three flags: $FAR : distance \geq 100[cm]$, $NEAR : 50[cm] < distance < 100[cm]$, and $VERY\_NEAR : distance \leq 50[cm]$. The contextual objects in the video scene (sofa, table, and kitchen) have been modelled in 3D.



(a) (b)

Fig. 4. Video sequences for elderly care at home application. Figures (a) and (b) respectively show the observed elderly man and woman.

All the attributes are automatically computed by a tracking method, which is able to compute the reliability measures of the attributes. These reliability measures account the quality and coherence of the acquired data.

The learning process applied over the 68000 frames have resulted in a hierarchy of 670 state concepts and 28884 event concepts. From the 670 states, 338 state concepts correspond to terminal states (50.4%). From the 28884 events, 1554 event concepts correspond to events occurring between terminal states (5.4%). This number of state and event concepts can be reduced considering a state stability parameter, defining the minimal duration for considering a state as stable. Two learned events are described below.

This evaluation consists in comparing the recognised events with the ground-truth of a sequence. Different 750 frames from the elderly woman video are used for comparison, corresponding to a duration of 1.33 minutes. The recognition process has obtained as result the events summarised in Figure 5.

The evaluation has obtained 5 true positives (TP) and 2 false positives (FP) on event recognition. This results in a precision (TP/(TP+FP)) of 71%. MILES has been able to recognise all the events from the ground-truth, but also has

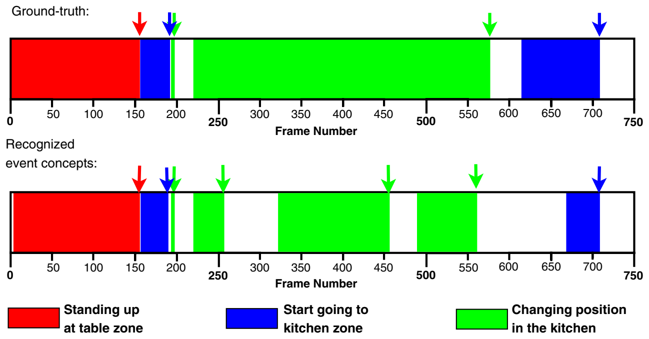
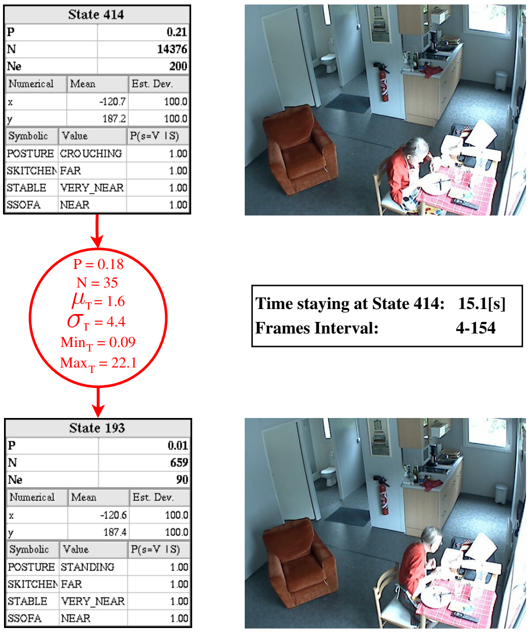


Fig. 5. Sequence of recognised events and ground-truth for the elderly woman video. The coloured arrows represent the events, while coloured zones represent the duration of a state before the occurrence of an event.



State 414		
P		**0.21**
N		**14376**
Ne		**200**
Numerical	Mean	Est. Dev.
x	-120.7	100.0
y	187.2	100.0
Symbolic	Value	P(s=V | S)
POSTURE	CROUCHING	1.00
SKITCHEN	FAR	1.00
STABLE	VERY_NEAR	1.00
SSOFA	NEAR	1.00

Time staying at State 414: 15.1[s]
Frames Interval: 4-154

State 193		
P		**0.01**
N		**659**
Ne		**90**
Numerical	Mean	Est. Dev.
x	-120.6	100.0
y	187.4	100.0
Symbolic	Value	P(s=V | S)
POSTURE	STANDING	1.00
SKITCHEN	FAR	1.00
STABLE	VERY_NEAR	1.00
SSOFA	NEAR	1.00

Fig. 6. Event *standing from the table zone*. On the left, the learned event is in red and the learned states are in black. The right-top image corresponds to a video frame before the event, while the right-bottom image corresponds to the frame which has caused the occurrence of the event. The black square contains the information about the starting state at the moment of occurrence of the event.

recognised two inexistent events, and has made a mean error on the starting state duration of 4 seconds. These errors are mostly due to bad segmentation near the kitchen zone, which had strong illumination changes, and to the similarity between the colours of the elderly woman legs and the floor. The results are encouraging considering the fact that the description of the sequence generated by a human has found a very close representation in the hierarchy.

Rich information about each learned state and event can be obtained with MILES. As an example, the learned event **Standing up at Table Zone** is explained in detail. This event has been detected when the elderly woman has begun to stand up from the chair. With the available pixel information it is not possible to say that the elderly woman was sitting on the chair, but just that she has changed her posture after a stable period being in a crouching posture. This event has been recognised 35 times, and it corresponds to 18% of possible events from the sitting state, as depicted in Figure 6. The results show that the system is able to learn and recognise meaningful events occurring in the scene. The computer time performance of MILES is 1300$[frames/second]$ for a video with one tracked object and six attributes, showing the real-time capability of the learning approach. However, the learned events are frequent and stable, but are not always meaningful for the user. Despite the calculation of the category utility, which formally measures the information density, an automatic process for measuring the usefulness of the learned events for the user is still needed.

5 Conclusions

A new event learning approach has been proposed, able to incrementally learn general events occurring in a video scene. The incremental nature of the event learning process is well suited for real world applications as it considers the incorporation of new arriving information with a minimal processing time cost. Incremental learning of events can be useful for abnormal event recognition and for serving as input for higher level event analysis. MILES allows to learn a model of the states and events occurring in the scene, when no a priori model is available. It has been conceived for learning state and event concepts in a general way. Depending on the availability of tracked object features, the possible combinations are large. MILES has shown its capability for recognising events, processing noisy image-level data with a minimal configuration effort. The proposed method computes the probability of transition between two states, similarly as HMM. The contribution MILES is to learn the global structure of the states and the events and to structure them in a hierarchy.

However, more evaluation is still needed for other type of scenes, for other attribute sets, and for different number and type of tracked objects. The anomaly detection capability of the approach on a large application must be also be evaluated. Future work will be also focused in the incorporation of attributes related to interactions between tracked objects (e.g. meeting someone). The automatic association between the learned events and semantical concepts and user defined events will be also studied.

References

1. Gennari, J., Langley, P., Fisher, D.: Models of incremental concept formation. In: Carbonell, J. (ed.) Machine Learning: Paradigms and Methods, pp. 11–61. MIT Press, Cambridge (1990)
2. Ghahramani, Z.: Learning dynamic bayesian networks. In: Giles, C.L., Gori, M. (eds.) IIASS-EMFCSC-School 1997. LNCS (LNAI), vol. 1387, pp. 168–197. Springer, Heidelberg (1998)
3. Hu, W., Tan, T., Wang, L., Maybank, S.: A survey on visual surveillance of object motion and behaviors. IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews 34(3), 334–352 (2004)
4. McKusick, K., Thompson, K.: Cobweb/3: A portable implementation. Technical report, Technical Report Number FIA-90-6-18-2, NASA Ames Research Center, Moffett Field, CA (September 1990)
5. Piciarelli, C., Foresti, G., Snidaro, L.: Trajectory clustering and its applications for video surveillance. In: Proceedings of the IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS 2005), pp. 40–45. IEEE Computer Society Press, Los Alamitos (2005)

6. Sridhar, M., Cohn, A., Hogg, D.: Learning functional object-categories from a relational spatio-temporal representation. In: Proceedings of the 18$^{\text{th}}$ European Conference on Artificial Intelligence (ECAI 2008), Patras, Greece, July 21-25, pp. 606–610 (2008)
7. Toshev, A., Brémond, F., Thonnat, M.: Unsupervised learning of scenario models in the context of video surveillance. In: Proceedings of the IEEE International Conference on Computer Vision Systems (ICCV 2006), January 2006, p. 10 (2006)
8. Vu, T., Brémond, F., Thonnat, M.: Automatic video interpretation: a novel algorithm for temporal scenario recognition. In: Proceedings of the 18$^{\text{th}}$ International Joint Conference on Artificial Intelligence (IJCAI03), Acapulco, Mexico (August 2003)
9. Xiang, T., Gong, S.: Video behavior profiling for anomaly detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(5), 893–908 (2008)