Hierarchical and Incremental Event Learning Approach based on Concept Formation Models

Marcos D Zúñiga[a], François Brémond[b], Monique Thonnat[b]

[a] *Electronics Department - UTFSM, Av. España 1680, Valparaíso, Chile*
[b] *INRIA - Projet PULSAR, 2004 rte. des Lucioles, Sophia Antipolis, France*

Abstract

We propose an event learning approach for video, based on concept formation models. This approach incrementally learns on-line a hierarchy of states and event by aggregating the attribute values of tracked objects in the scene. The model can aggregate both numerical and symbolic values.

The utilisation of symbolic attributes gives high flexibility to the approach. The approach also proposes the integration of attributes as a doublet value-reliability, for considering the effect in the event learning process of the uncertainty inherited from previous phases of the video analysis process.

Simultaneously, the approach recognises the states and events of the tracked objects, giving a multi-level description the object situation.

The approach has been evaluated for an elderly care application and a rat behaviour analysis application. The results show that the approach is capable of learning and recognising meaningful events occurring in the scene, and to build a rich model of the objects behaviour. The results also show that the approach can give 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.

Keywords: incremental event learning, hierarchical event model, human behaviour, reliability measures, symbolic attribute

1. Introduction

Video event learning presents relevant applications related to abnormal behaviour detection, as elderly health care [19], [11], and traffic monitoring [8]. In this sense, the utilisation of incremental models for event learning should be the natural step further real-time applications for handling unexpected events. Apart from being well-suited for real-time applications because of the inexpensive learning process, this incremental characteristic learning allows the systems to easily adapt their response to different situations. Also, the dependence on enormous datasets for each particular application is reduced.

The focus of this work is in applications for incremental event learning, where several objects of diverse type can interact in the scene (e.g. persons, vehicles). The events of interest are also diverse (e.g. events related to trajectories, human posture) as the focus of interest is learning events in general. The objects simultaneously evolving in the scene can be many, but the interest is centred in objects which can be individually tracked in order to be able of recognising the events each object is participating.

We propose 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 **states** and **events**. Reliability measures are used to focus the learning process on the most valuable information. Simultaneously, the approach **recognises** new occurrences of **states** and **events** previously learnt. The only hypothesis of the approach is the availability of tracked object attributes, which are the needed input for the approach. This approach is able to learn **states and events in general**, so **no limitation** is imposed on the **nature** or **number** of attributes to be utilised in the learning process.

As previously described, the hierarchical model of the proposed approach can be incrementally updated. This feature is based on **incremental concept formation models** [4]. These concept formation models evaluate the goodness of the concepts represented by the formed clusters in a hierarchical model, with the added constraint that learning must be incremental. The main contributions of the proposed learning approach, with respect to incremental concept formation models, are:

- The capability of the hierarchical model to **learn events**, as an explicit transition between two states (described in Sections 3.1 and 4.4).
- The utilisation of **reliability measures** for weighting the

Email addresses: marcos.zuniga@usm.cl (Marcos D Zúñiga), Francois.Bremond@sophia.inria.fr (François Brémond), Monique.Thonnat@sophia.inria.fr (Monique Thonnat)

URL: http://profesores.elo.utfsm.cl/~mzuniga (Marcos D Zúñiga), http://www-sop.inria.fr/teams/pulsar/personnel/Francois.Bremond (François Brémond), http://www-sop.inria.fr/teams/pulsar/personnel/Monique.Thonnat (Monique Thonnat)

contribution of data according to their quality, as a way to focus learning on meaningful information (for details, see Section 4.3).

- The extended utilisation of the concept of **acuity** to represent different **normalisation** scales and units associated to different attributes, and also represent the interest of users for different applications (see Section 3.2, for details).

- The incorporation of the **acuity** to the **numerical category utility**, in order to balance the contribution of numerical and symbolic attributes to the category utility. (see Section 4.2).

In a step further to bridge the gap between image-level data and high-level semantic information, this work extends previous work presented in [21] and [22] by integrating symbolic attribute information to the hierarchical model in a way that both **numerical** and **symbolic** attribute values can be in a common state model. The utilisation of symbolic attributes gives high flexibility to the approach, allowing the user to add significantly semantic attributes for assisting on scene interpretation.

Also, the approach can simultaneously learn different hierarchies representing different learning contexts (i.e. different states and events of interest). We propose a general representation for the context of each learning process and extend the analysis of each involved process for an easier implementation. The source code of the algorithm is publicly available [1].

The approach has been extensively verified over both simulated and real data-sets. The real data-sets has been utilised for specific events for home-care (e.g. approaching to a table, crouching) and rat behaviour learning (position and velocity events).

This paper is organised as follows. In Section 2 the state-of-the-art on incremental event learning approaches is presented. Section 3 describes the proposed event learning approach in general, and Section 4 focuses on describing the learning process in detail. Finally, Section 5 presents the experiments performed on simulated and real data-sets.

2. State-of-the-Art

Most of video event learning approaches for abnormal behaviour recognition are supervised, requesting annotated videos representative of the events to be learnt in a training phase [7], [6], [2]. As well described in [17], these approaches normally use general techniques as Hidden Markov Models (HMM) [13]. Some authors use hierarchical models, as they facilitate learning and generalisation. HMMs are robust, but require hierarchical (HHMM) and time-duration modelling for representing events with varying temporal and spatial scales, increasing the complexity of these approaches.

Generalisation is one of the keys to simplify the process of semantic interpretation. In [10], the authors propose an approach for abnormal behaviour detection, using unsupervised learning for two hierarchical representations, one for description of the observation and the other for temporal description. In [15], the authors proposed a fall detection algorithm that uses HHMM, hand designed and operating on an observation sequence of rectified angles.

Few approaches can learn events in an unsupervised way using clustering techniques. For instance, [18] use the clusters of attributes obtained with a Gaussian Mixture Model to represent the states of an HMM, [14] learn events using spatial relationships between objects in an unsupervised way, but performed off-line, and [16] apply unsupervised learning of composite events using the APRIORI clustering algorithm. 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, [12] propose a method for incremental trajectory clustering by mapping the trajectories into the ground plane decomposed in a zone partition. Their approach performs learning only on spatial information, it cannot take into account time information, and do not handle noisy data.

In conclusion, few work has been found on hierarchical and incremental approaches for abnormal behaviour detection. A critical aspect not considered in the current approaches is the uncertainty of mobile object attributes present in real applications and how this uncertainty can affect the model construction.

Following these directions, the current work is based on *incremental concept formation models* [4]. 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 [9].

The proposed approach takes profit of this hierarchical structure, extending it to represent events, incorporate the effect of uncertainty in data, and to manage symbolic attributes which facilitate semantic interpretation.

3. Incremental state and event learning approach

As previously stated, the proposed approach is an extension of **incremental concept formation models** [4, 1] 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 the approach is the availability of tracked object attributes (e.g. position, posture, class, speed).

The proposed approach has been called *MILES*, acronym standing for **M**ethod for **I**ncremental **L**earning of **E**vents and **S**tates. The approach has received its name since its first version, presented in [21]. MILES state hierarchy construction is mostly based on COBWEB [3] algorithm, but also considering ideas from other existing incremental concept formation approaches, as CLASSIT [4] algorithm.

3.1. The hierarchy of states and events

MILES builds a **hierarchy** of state and event concepts **H**, based on the **state and event instances** extracted on-line from

[1] The algorithm has been developed with C++, using QT libraries, and is available at `http://profesores.elo.utfsm.cl/~mzuniga/MILES.zip`

the tracked object attributes. It is desirable (but not necessary) that the input data contains an estimate of the reliability on information. This hierarchy is formed by two building blocks:

State concept: It is the modelling of a state, as previously defined. A **state concept** $S^{(c)}$, in a hierarchy **H**, is modelled as:

- its **number of occurrences** $N(S^{(c)})$ and its **probability of occurrence** $\mathcal{P}(S^{(c)}) = N(S^{(c)})/N(S^{(p)})$. ($S^{(p)}$ is the root state concept of **H**),
- the **number of event occurrences** $N_E(S^{(c)})$, corresponding to the number of times that the state $S^{(c)}$ passed to another state, generating an event.
- a **set of numerical 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 $N_i \sim \mathcal{N}(\mu_{n_i}; \sigma_{n_i})$,
- a **set of symbolic attribute models** $\{s_j\}$, with $j \in \{1, .., S\}$, where s_j is represented by every possible value for the attribute, and conditional probabilities $P(V_{s_j}^{(k)}|S^{(c)})$ representing the frequency of occurrence of a the k-th value $V_{s_j}^{(k)}$ for s_j, given $S^{(c)}$.

Event concept: It is the modelling of the transition between two state concepts. 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 **H**. An **event concept** $E^{(c)}$, in a hierarchy **H**, is modelled as:

- 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. In the hierarchy, an event concept is represented as a unidirectional link between two state concepts. 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.

3.2. The Learning Contexts

The learning process associated to a particular hierarchy **H** is guided by a **learning context Z**. A learning context corresponds to the description of a particular scope of the events of interest for the user. Multiple learning contexts can be defined and simultaneously processed, according to user interests. Each learning context requires the definition of:

- the moving object classes involved in the particular learning process, defining a list of the object classes of interest or stating that **any** class is of interest.
- the attributes of interest (numerical or symbolic). Normally, there is an intermediate step for obtaining these attributes from involved objects, as these attributes can be derived from other object attributes (e.g. symbolic attribute defining a zone in the scene, derived from object position).
- Particularly, for each **numerical attribute** of interest n_i, a normalisation value A_{n_i} must be also defined. A_{n_i} represents the lower bound for the numerical attribute change to be considered as meaningful. In other words, the difference between the mean value for a numerical attribute n and the value of the attribute for a new instance will be considered as significant and noticeable when this difference is higher than A_{n_i}.

  This normalisation value corresponds to the concept of **acuity**, utilised by [4] and described as a system parameter that specifies the minimum value for attributes σ in the CLASSIT algorithm for incremental concept formation. In psycho-physics, 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 CLASSIT is that the **acuity** was used as a single parameter, while A_{n_i} acuity values are defined for each numerical attribute to be learnt for a given context. This improvement allows to represent the different normalisation scales and units associated to different attributes, and can also represent the interest of users for different applications. For instance, a trajectory position attribute x could have an acuity of 50 *centimetres* for an application with a camera in an office environment, while for the same attribute, the acuity could be *two metres* for a parking lot application with a camera far from the objects, where the user is not interested in little details on position change.
- In particular, for each **symbolic attribute** s_j, it is necessary to list the associated values of interest.

As an example, for a *Position-Posture* learning context, as shown in Figure 2, the user can be interested in learning the events associated to a Person position (x, y), together with the human posture in an office environment. As an office is a small closed area, appropriate normalisation values for position attributes can be *50 centimetres*. Then, this context mixes numerical position attribute information, with symbolic posture attribute information.

```
Learning Context  Position_Posture {
        Involved Objects:  Person
        Attributes:
                Numerical  x : 50 [cm]
                Numerical  y : 50 [cm]
                Symbolic  Posture : { Standing, Crouching, Sitting, Lying }
}
```

Figure 2: Definition of a Position-Posture learning context for Person class in an office environment.

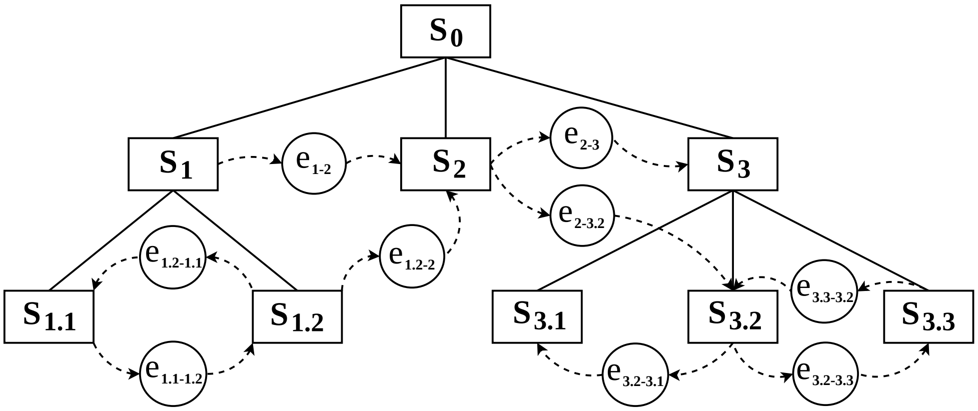


Figure 1: Example of a hierarchical event structure resulting from the proposed event learning approach. Rectangles represent states, while circles represent events.

It is worthy to notice that the purpose of learning contexts is to increase the possibilities of the users to customise the learning process according to the information of interest to an application. In other words, nothing limits a user to define a learning context with all the available attributes. All these possibilities of customisation by the user, give a high flexibility to the proposed approach for adapting to a wide variety of applications and typical issues present in the video understanding domain. Also, symbolic attributes allow the user to define attributes which help in the semantic interpretation, bridging the gap between image-level data and high-level information.

3.3. Contextualised Objects and State Instances

According to the learning context, pertinent attributes of a tracked object have to be extracted (or generated). In the context of MILES, each mobile object must also store information related to their position in the hierarchy tree, for each learning context in which it participates. Then, a contextualised object **o** will be an extended representation of a tracked object. This object **o**, for each learning context Z it participates, must then contain:

- a **state instance**, which is an instantiation of a state concept, associated to the object **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 $i \in \{1, \ldots, T' + S'\}$, where R_i is the reliability measure associated to the obtained value V_i for the attribute v_i. T' and S' are the number of pertinent numerical and symbolic attributes, respectively, according to learning context Z. The measure $R_i \in [0, 1]$ is 1 if associated data is totally reliable, and 0 if totally unreliable, allowing to control the learning process according to quality of information. Attribute v_i can be numerical or symbolic.

- For each level in the hierarchy H, associated to Z:

  - Last detected **event concept** $E^{(c)}$ for object **o**.
  - Previously detected **state concept** $S_a^{(c)}$. Corresponds to a matching between **state concept** $S_a^{(c)}$ and a **state instance** $S^{(o)}$ previously extracted from object **o**.
  - Currently detected **state concept** $S_b^{(c)}$. Corresponds to a matching between **state concept** $S_b^{(c)}$ and the **state instance** $S^{(o)}$ currently extracted from object **o**.

Now, with all these elements and their interactions properly described, details on the event learning process can be presented in next Section 4.

4. MILES Learning Process

MILES needs that the objects are tracked in order to detect the occurrence of *events*. There is no constraint on the number and nature of attributes, as MILES has been conceived for learning state and event concepts in general, as discussed in section 3.2.

Initially, before the first execution of MILES, and for each defined learning context **Z**, a hierarchy **H** is initialised as an empty tree. If MILES has been previously executed, the incremental nature of MILES learning process allows that the hierarchy **H** resulting from this previous execution can be utilised as the initial hierarchy of a new one.

The input of MILES corresponds to a list of contextualised mobile objects **O**, according to the defined learning contexts. At each video frame, MILES utilises **O** for updating each hierarchy **H**. Considering a particular learning context **Z** and its corresponding hierarchy **H**, MILES first gets the set of triplets $\mathbf{T_o}$, equivalent to a **state instance**(see section 3.3), for each object **o** in **O**, pertinent to **Z**. These triplets will be the input for the state concept updating process of **H**. This updating process is described in Section 4.1. 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 learning context for which an object **o** in **O** is pertinent.

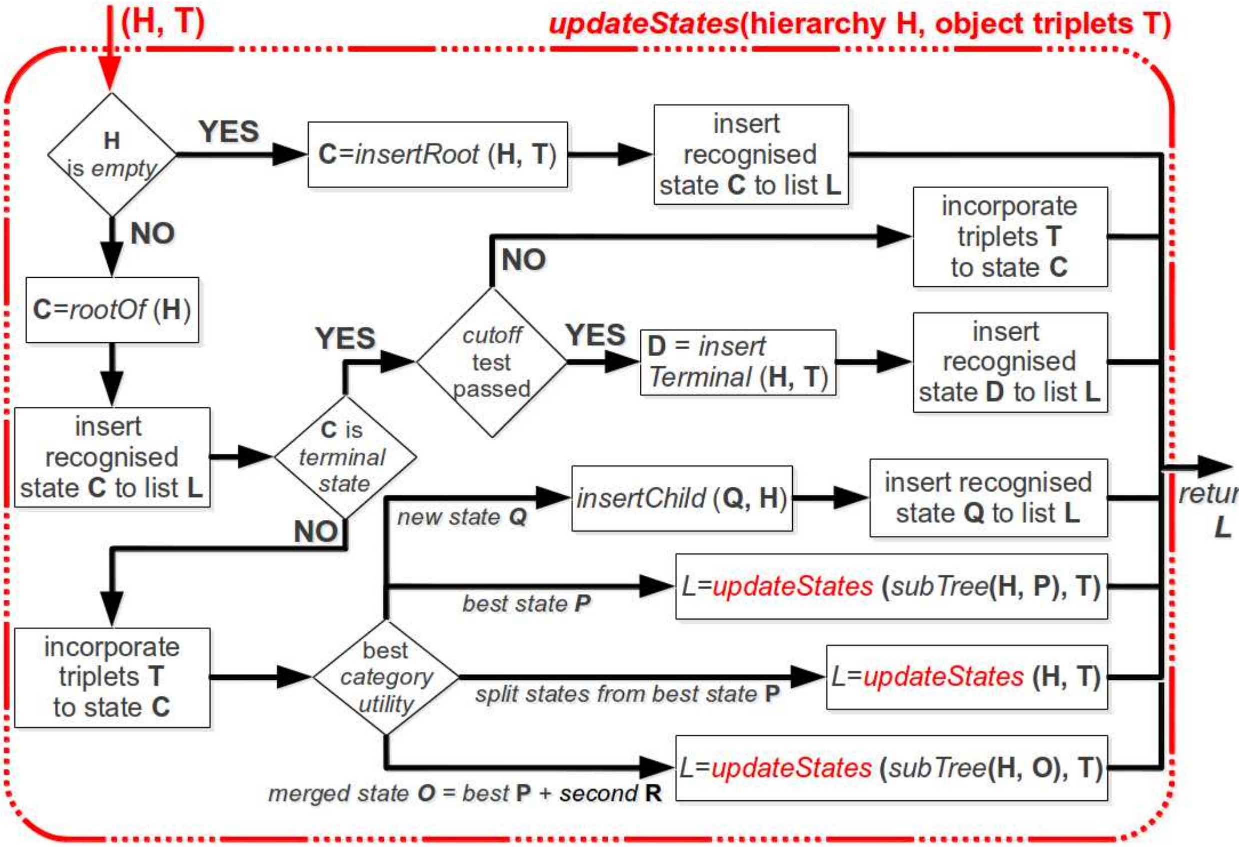

(H, T)
updateStates(hierarchy H, object triplets T)
H is empty
YES
NO
C=insertRoot (H, T)
insert recognised state C to list L
incorporate triplets T to state C
C=rootOf (H)
insert recognised state C to list L
C is terminal state
YES
NO
cutoff test passed
NO
YES
D = insert Terminal (H, T)
insert recognised state D to list L
new state Q
insertChild (Q, H)
insert recognised state Q to list L
return L
best state P
L=updateStates (subTree(H, P), T)
incorporate triplets T to state C
best category utility
split states from best state P
L=updateStates (H, T)
merged state O = best P + second R
L=updateStates (subTree(H, O), T)

Figure 3: Scheme of the state concept updating algorithm.

4.1. States Updating Algorithm

State concept updating is a recursive process, as depicted in Figure 3.

The algorithm starts by accessing the analysed state **C** from hierarchy **H** (with *rootOf* returning the root state of **H**). Notice that, in the context of the algorithm, a hierarchy not necessarily corresponds to the complete tree, as the algorithm recursively utilises sub-branches of the hierarchy. The initialisation of **H** is performed by creating a state with the triplets **T**, for the first processed object. Remember that **T** represents the **state instance** for object **o**, given a learning context **Z**.

Then, for the case that **C** corresponds to a terminal state (state with no children), a *cutoff* test is performed. The **cutoff** is a criteria utilised for stopping the creation (i.e. specialisation) of children states. It is defined in

$$\text{cutoff} = \begin{cases} \textbf{true} \;\; \text{if} & \{\mu_{n_i}^{(C)} - V_{n_i} \leq A_{n_i} | \forall i \in \{1,..,T'\}\} \\ & \wedge \;\; \{\mathcal{P}(V_{s_j}|s_j{}^{(C)}) = 1 | \forall j \in \{1,..,S'\}\} \\ \textbf{false} \;\; \text{else} & \end{cases} , \quad (1)$$

where V_{n_i} is the value of a numerical attribute n_i, and V_{s_j} is the value of the symbolic attribute s_j. $\mu_{n_i}^{(C)}$ is the mean value of n_i for **C**. $\mathcal{P}(V_{s_j}|s_j{}^{(C)})$ is the conditional probability of the value V_{s_j}, given s_j of **C**. T' and S' are the number of pertinent numerical and symbolic attributes for **Z**, respectively.

This equation means that the learning process will stop at **C** if no meaningful difference exists between a numerical attribute value at **T** and the mean value of the attribute for **C** (using **acuity** A_{n_i} criteria), or if every symbolic attribute value in **T** is totally represented in **C** (probability equal to one for the attribute value). This means that the learning process will stop if **no noticeable difference between the attribute values is found**.

If the *cutoff* test is passed (noticeable difference found), the function *insertTerminal* generates two children for **C**, one initialised with **T** and the other as a copy of **C**. Then, **T** is incorporated to **C** (process described in Section 4.3). In this terminal state case, the updating process then stops.

If **C** has children, first **T** is immediately incorporated to **C**. In order to determine in which state concept the triplets list **T** is next incorporated (i.e. the state concept is recognised), a quality measure for state concepts called **category utility** (CU) is utilised, which is discussed in detail on Section 4.2. Then, the different alternatives for the incorporation of **T** are:

1. Incorporating **T** to an existing state **P** gives the best CU score. In this case, *updateStates* is recursively called, considering **P** as root.
2. The generation of a new state concept **Q** from **T** gives the best CU score. In this case, **Q** is inserted as child of **C**, and the updating process stops.
3. Consider **M** as the resulting state from merging the best state **P** and the second best state **R**. Also, consider **y** as the CU score of replacing **P** and **R** with **M**. If **y** is the best score, **H** is modified by the **merge operator**. Then,

updateStates is recursively called, using the sub-tree from state **M** as the tree to be analysed. The **merge operator** is described in detail in Section 4.5.

4. Consider **z** as the *CU* score of replacing state **P** with its children. If **z** is the best score, **H** is modified by the **split operator**. This process implies to suppress the state concept **P** together with all the events in which the state is involved, as depicted in Figure 4. Then, *updateStates* is called, using the sub-tree from the current state **C** again.

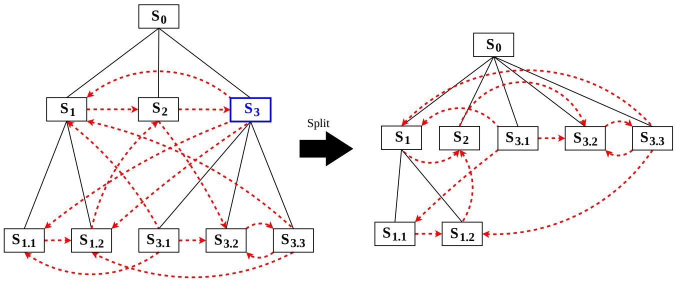


Figure 4: Split operator in MILES algorithm. The blue box represents the state to be split. Red dashed lines represent events. Notice that the split operator suppresses the state S_3 and its arriving and leaving events, and ascends the children of S_3 in the hierarchy.

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*.

4.2. The Category Utility

As previously discussed, the **category utility** measures how well the **state instances** are represented by a given **state concept**. This function has been derived by Gluck and Corter [5]. Category utility attempts to maximise intra-class similarity and inter-class differences, and it also provides a principled trade-off between predictiveness and predictability [3]. A measure similar to the category utility function from COBWEB/3 [9] algorithm has been considered.

For the set of numerical attributes, the numerical category utility $CU_k(num)$, for a given state concept S_k, is defined as:

$$CU_k(num) = \frac{\mathcal{P}(S_k)\sum_{i=1}^{T'}\left(\frac{A_{n_i}}{\sigma_{n_i}^{(k)}} - \frac{A_{n_i}}{\sigma_{n_i}^{(p)}}\right)}{2\cdot T'\cdot\sqrt{\pi}}, \qquad (2)$$

where $\sigma_{n_i}^{(k)}$ is the standard deviation for the numerical attribute n_i in S_k, and $\sigma_{n_i}^{(p)}$ is the standard deviation for n_i in the parent or root node S_p. The value A_{n_i} corresponds to the **acuity** for n_i.

The incorporation of the acuity term A_{n_i} to the equation 2 establishes a difference with the preceding versions of numerical category utility in the state-of-the-art. This is done to balance the contribution of numerical and symbolic attributes to the category utility. The obtained attribute contribution value always belongs to the interval $[0, 1]$, as A_{n_i} is the lower bound for $\sigma_{n_i}^{(k)}$. Also, the acuity is useful to normalise the contributions of numerical attributes representing different metric units (e.g. position and velocity) and scales (e.g. a position in metres and a distance in centimetres).

For the set of symbolic attributes, the symbolic category utility $CU_k(sym)$, for S_k, is defined as:

$$CU_k(sym) = \frac{\mathcal{P}(S_k)\sum_{i=1}^{S'}\sum_{j=1}^{J_i}\left(\mathcal{P}(s_i = V_{s_i}^{(j)}|S_k)^2 - \mathcal{P}(s_i = V_{s_i}^{(j)}|S_p)^2\right)}{S'}, \qquad (3)$$

where $\mathcal{P}(s_i = V_{s_i}^{(j)}|S_k)$ is the conditional probability that the symbolic attribute s_i has a value $V_{s_i}^{(j)}$ in S_k, while $\mathcal{P}(s_i = V_{s_i}^{(j)}|S_p)$ is the conditional probability that s_i has a value $V_{s_i}^{(j)}$, in the parent or root node S_p.

Then, for a set of mixed symbolic and numerical attributes, the overall category utility CU_k, given a state concept S_k, is the sum of the contributions of both sets of features:

$$CU_k = CU_k(sym) + CU_k(num). \qquad (4)$$

Finally, the category utility CU for a class partition of K state concepts is defined as:

$$CU = \sum_{k=1}^{K}\frac{CU_k}{K} \qquad (5)$$

4.3. Incorporation of New Object Attribute Values

Upon the arrival of a new **state instance**, the attribute information of the instance must be used to update the state and event concept information. According to the type of attribute the information updating process differs.

For the case of a numerical attribute n, the information about the mean value μ_n and the standard deviation σ_n must be updated. The proposed updating functions are incremental in order to improve the processing time performance of the approach. For μ_n, the function is presented in Equation (6).

$$\mu_n(i) = \frac{V_n\cdot R_n + \mu_n(i-1)\cdot Sum_n(i-1)}{Sum_n(i)}, \qquad (6)$$

with

$$Sum_n(i) = R_n + Sum_n(i-1), \qquad (7)$$

where V_n is the value in the new instance for n and R_n corresponds to its reliability. Hence, the reliability R_n weights the contribution of V_n to μ_n. Sum_n is the accumulation of reliability values R_n for n.

The updating function for σ_n is presented in Equation (8).

$$\sigma_n(i) = \sqrt{\frac{Sum_n(i-1)}{Sum_n(i)}\cdot\left(\sigma_n(i-1)^2 + \frac{R_n\cdot(V_n - \mu_n(i-1))^2}{Sum_n(i)}\right)}. \qquad (8)$$

In the case that a new state concept is generated from the attribute information of the instance, the initial values taken for Equations (6), (7), and (8) with $i = 0$ 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, as defined in Section 3.2.

In case that, after updating $\sigma_n(i)$, its value is lower than the *acuity* A_n, $\sigma_n(i)$ becomes equal to A_n. This way, the acuity value establishes a lower bound for the standard deviation, avoiding the possibility of zero division.

For a symbolic attribute s it is necessary to update the conditional probability $\mathcal{P}(s = V_s^{(j)}|S)$ of each possible value $V_s^{(j)}$ of s, given S. For this purpose, reliability measures R_s are utilised in order to weight the quality of new incoming information, as presented in Equations (9), (10), and (11).

$$\mathcal{P}(s = V_s^{(j)}|S)[i] = \begin{cases} \dfrac{Sum_{V_s}^{(j)}(i)}{Sum_s(i)} & if \quad V_s = V_s^{(j)} \\[2ex] \dfrac{Sum_{V_s}^{(j)}(i-1)}{Sum_s(i)} & else \end{cases} \qquad (9)$$

with

$$Sum_{V_s}^{(j)}(i) = R_s + Sum_{V_s}^{(j)}(i-1), \qquad (10)$$

and

$$Sum_s(i) = R_s + Sum_s(i-1), \qquad (11)$$

where V_s is the value in the new instance for s, and R_s corresponds to its reliability. $V_s^{(j)}$ is the j-th possible value s. The functions $Sum_{V_s}^{(j)}(i)$ correspond to the accumulated reliability for each s value V_s, while the function $Sum_s(i)$ is the overall accumulated reliability for s.

4.4. Events Updating Algorithm

After the states updating phase (see Section 4.1). the changes of **state concept** occurred for an object **o** must update the events information according to the change of state. The occurrence of a state transition updates all the events representing the combinations between the analysed state concept from the stored list, where the possible combinations are:

- All the states of a lower level in the new list, if the state at its same level in the new list is different than the analysed state.
- The state at its same level in the new list if it is different than the analysed state.
- All the states at a higher level in the new list which do not have a *kinship relation* (defined below) with the analysed state.

A ***kinship relation*** *between two states S_m and S_n in the hierarchy exists if S_m is (directly or indirectly) the ascendant or one of the descendants of the state S_n in the hierarchy. This means that the one state is related to the other as parent, or son, or grand-parent, or grand-son, and so on.*

Examples of these state combinations can be found in Figure 5.

If an event E corresponds to a first detected event, a new event representation is created and associated to the generating state S_a and the arriving state S_b.

Then, the updated list of current states at different levels in the hierarchy is utilised to update the current states information of the object **o**.

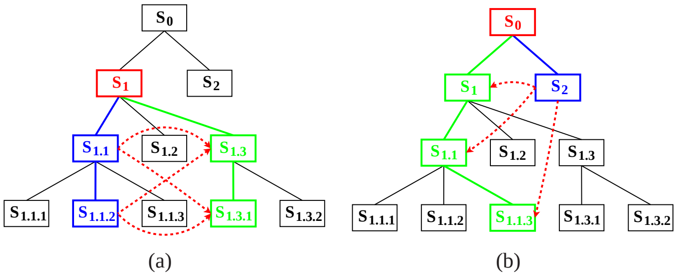


Figure 5: Examples of list comparisons for determining the events to update. Blue elements represent the previously stored states for a tracked object. Green elements represent the updated states obtained with the function *updateStates*. The red box represents the state concept which is common to both lists. The dashed red lines represent the events to update for two different cases (a) and (b).

4.5. Merge Operator

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 S_M, as depicted in Figure 6.

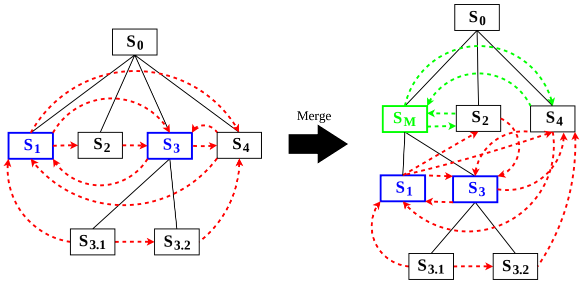


Figure 6: Merging states and events in MILES algorithm. Blue boxes represent the states to be merged, and the green box represents the resulting merged state. Red dashed lines represent events, while the green dashed lines are the new events appearing from the merging process.

In order to generate the state S_M several considerations must be made:

- $N(S_M) = N(S_p) + N(S_q)$.
- $\mathcal{P}(S_M) = N(S_M)/N(S_r)$, with S_r the root node of the hierarchy.
- $N_E(S_M)$ corresponds to the number of events E having a starting state $S_a(E) = S_p$ or S_q, and as an ending state $S_b(E)$ a state not having a *kinship relation* with S_M.
- Each numerical attribute n_M for S_M can be updated using the Equations (12), and (13) for mean and standard deviation of n_M, respectively.

$$\mu_{n_M} = \frac{Sum_{n_p} \cdot \mu_{n_p} + Sum_{n_q} \cdot \mu_{n_q}}{Sum_{n_p} + Sum_{n_q}}, \qquad (12)$$

$$\sigma_{n_M}^2 = \frac{Sum_{n_p} \cdot (\Delta_{Mp}^2 + \sigma_{n_p}^2) + Sum_{n_q} \cdot (\Delta_{Mq}^2 + \sigma_{n_q}^2)}{Sum_{n_p} + Sum_{n_q}}, \qquad (13)$$

where Sum_{n_p} and Sum_{n_q} correspond to the accumulated reliability values for numerical attributes n_p and n_q, respectively. $\Delta_{Mp} = (\mu_{n_M} - \mu_{n_p})$ and $\Delta_{Mq} = (\mu_{n_M} - \mu_{n_q})$ were added to adjust the value of σ_{n_M}, considering the drift between the new mean μ_{n_M}, and the mean values μ_{n_p} and μ_{n_q}.

- Each symbolic attribute s_M for S_M can be updated using the Equation (14), for the conditional probability $\mathcal{P}(s_M)^{(j)}$, for the j-th value of the symbolic attribute s_M.

$$\mathcal{P}(s_M = V^{(j)}_{s_M} | S_M)[i] = \frac{Sum^{(j)}_{V_{s_p}} + Sum^{(j)}_{V_{s_q}}}{Sum_{s_p} + Sum_{s_q}}, \qquad (14)$$

where $Sum^{(j)}_{V_{s_p}}$ and $Sum^{(j)}_{V_{s_q}}$ correspond to the accumulated reliability values of the j-th value for symbolic attribute s_p and s_q, respectively. In the same way, Sum_{s_p} and Sum_{s_q} are the overall reliability values accumulation for s_p and s_q, respectively.

The last task for the merging operator is to represent the events incoming and leaving states S_p and S_q (green dashed lines in Figure 6) by generating new events which generalise the transitions as the events incoming and leaving the state S_M. For the **incoming events** to these states the event merge process is described as follows:

- If a state S_n is the starting state for an event $E_{n \to x}$ arriving to only one state S_x of the merging states S_p and S_q (as event $E_{S_2 \to S_3}$ between states S_2 and S_3 in Figure 6), a new event $E_{n \to M}$ must be generated with the same information as event $E_{n \to x}$, except for the arriving state that becomes the state S_M.

- If a state S_n is the starting state for the events $E_{n \to p}$ and $E_{n \to q}$ arriving to both states S_p and S_q (as events $E_{S_4 \to S_1}$ and $E_{S_4 \to S_3}$ in Figure 6), a new event $E_{n \to M}$ must be generated with:
  - $N(E_{n \to M}) = N(E_{n \to p}) + N(E_{n \to q})$
  - $\mathcal{P}(E_{n \to M}) = N(E_{n \to M}) / N_E(S_n)$.

Finally, **events leaving** the states S_p and S_q must be merged, with:

- $N(E_{M \to n}) = N(E_{p \to n}) + N(E_{q \to n})$
- $\mathcal{P}(E_{M \to n}) = N(E_{M \to n}) / N_E(S_M)$

5. Experiments and Results

5.1. *Illustration of MILES State and Event Representation*

In order to better understand the learning process, an illustration example is presented in this section. The example consists in ten persons evolving in a metro scene, starting at different positions and time instants. A top view of the scene is depicted in Figure 7. The evolution of the persons in the scene is represented by ten hand-crafted trajectories (T0 - T9) of eight coordinate points (x,y) in the ground plane of the scene.

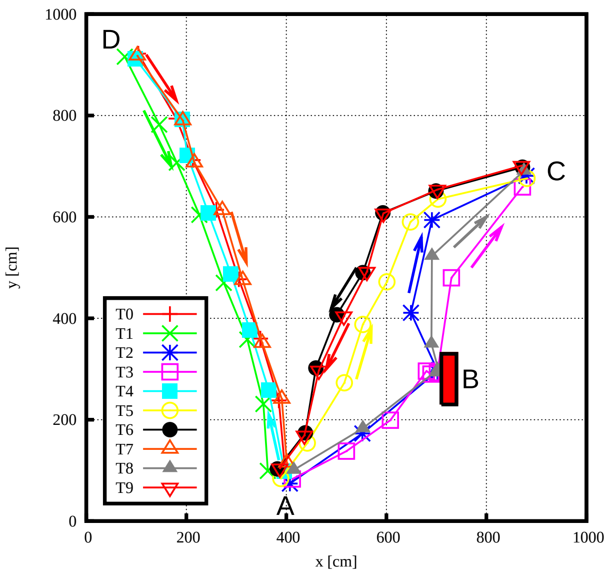


Figure 7: Top view of the metro scene illustration example. The ten hand-crafted trajectories (T0-T9) are displayed.

The scene consists of three Access/Exit zones (referenced in the Figure 7 as **A**, **C** and **D**), and a ticket vending machine zone **B**, represented as a red box in Figure 7. The ten persons evolve in the scene over 13 time instants.

The idea is to utilise a simple learning context consisting in the (x,y) person positions, with an acuity of $200[cm]$. Then, the evolution of the hierarchy of states and events in time can be analysed to understand the event learning process. Also, the relations between the obtained states and events and the trajectories of the persons can be studied to understand how the hierarchy represents the situations occurring in this scene.

Learning up to Time instant 1:

At this instant two persons (represented by T0 and T1) arrive from the zone **D** and two other persons (represented by T2 and T3) arrive from the zone **A**. This situation is represented by two different states of the hierarchy, because the person positions entering at the two different zones were similar enough to be represented in the same state concept. The positions of T0 and T1 are then represented by the State 1, while the positions of T2 and T3 by the State 2.

Figure 8(a) shows a top view of the scene where these the two new states are represented. Figure 8(b) depicts the maximal marginal probability for each point in the scene, given the current two states of the hierarchy.

Learning up to Time instant 3:

The evolution of the hierarchy until this instant is depicted in Figure 9. T4 starts walking in the direction of the zone **D**, while T5 goes in the direction of the zone **C**. The position of T4 and T5 is not different enough yet to generate a new state.

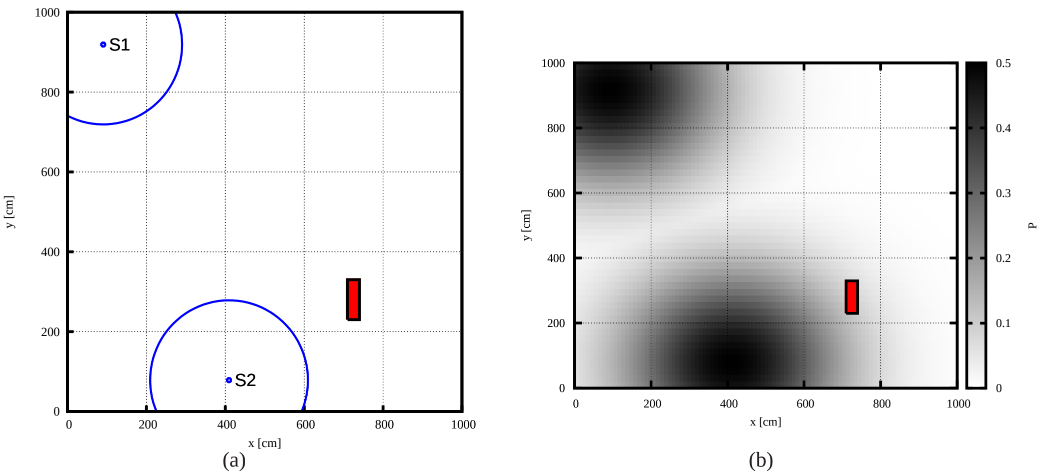


Figure 8: Hierarchy at instant 1. (a) Terminal states position in a top view of the scene. The oval surrounding the mean position of a state represents the standard deviation of this position. A state in the first level of the hierarchy is represented in blue. (b) Maximal marginal probability of a state. A darker colour represents a higher probability.

Then the probability of the State 2 is still reinforced. T0 and T1 walk in the direction of the zone **A**, but their position is similar enough to the position represented in the State 1, reinforcing its probability. Also, T7 arrives from the zone **D**, reinforcing the probability of the State 1 even more.

T2 and T3 walk to the ticket vending machine **B**. Now, their position is different enough to the one represented by the State 2, to induce the creation of two children states. One state (State 3) represents the position near the zone **A**, and the other represents the new created State 4 near the zone **B**. The new positions of T2 and T3 have also induced a change of state, represented by the first event in the hierarchy between States 3 and 4. This event is depicted in Figure 9, and graphically represented by an arrow between States 3 and 4, in Figure10(a).

Notice in Figure 10(b) that the new created state does not have a strong probability, compared with the other states of the hierarchy.

Learning up to Time instant 5:

The new position of T4 produces an adjustment of the position of State 8, while the new position of T5 induces the creation of a new event between States 8 and 9, as depicted in Figure 11(a). T5 walks in the direction of zone **C**. Then, the transition between States 8 and 9 seems imprecise, but this is one of the costs of considering a coarse value for the acuity of position attributes x and y. Also, T9 arrives to the scene from the zone **C**, reinforcing the probability of State 10.

Notice in Figure 11(b) that the permanence of T2 and T3 at the zone **B** has reinforced the probability of the State S9 near this zone. Also notice that the reposition of State 8, induced by person T4, has also reinforced the probability of occurrence of the State 8.

Learning up to Time instant 7:

At this time instant, the hierarchy has arrived to a stable number of states. The new position of T6 induces a new event between States 12 and 9. At the same time, the position of T2 induces a new event between States 9 and 12 (in that order), as depicted in Figure 12(a). Figure 12(b) shows that even the probability map has arrived to a quite stable state, where only slight differences can be observed.

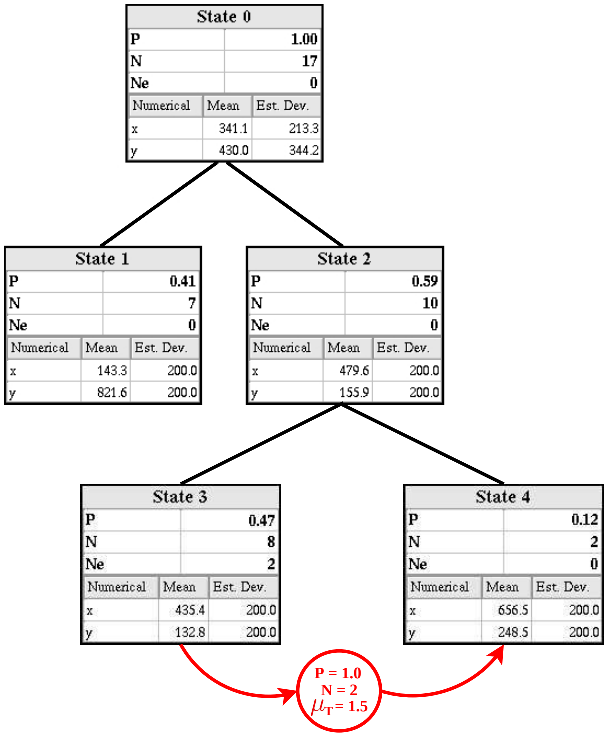
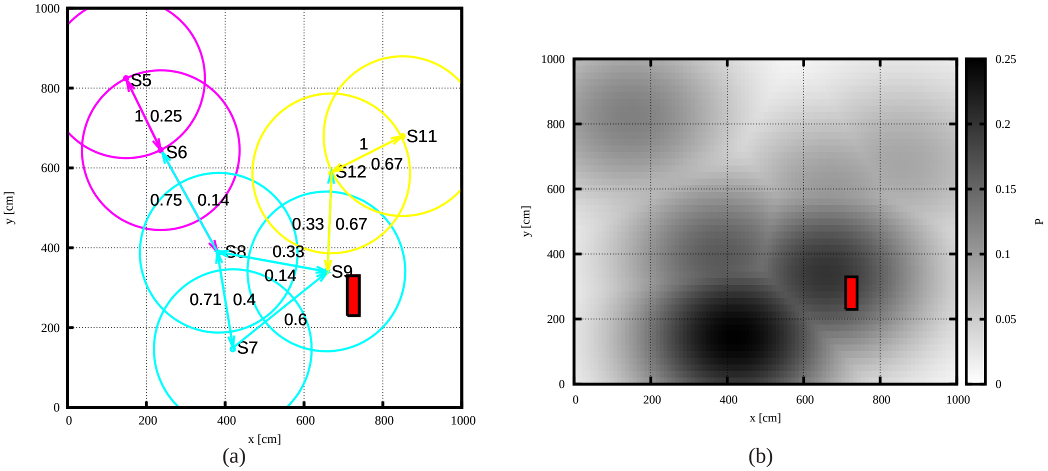


Figure 9: Hierarchy obtained up to instant 3. Events are coloured in red.

From this time instant and until the end of the illustration example, the hierarchy is very stable, only showing some new events and updates in the states probability.

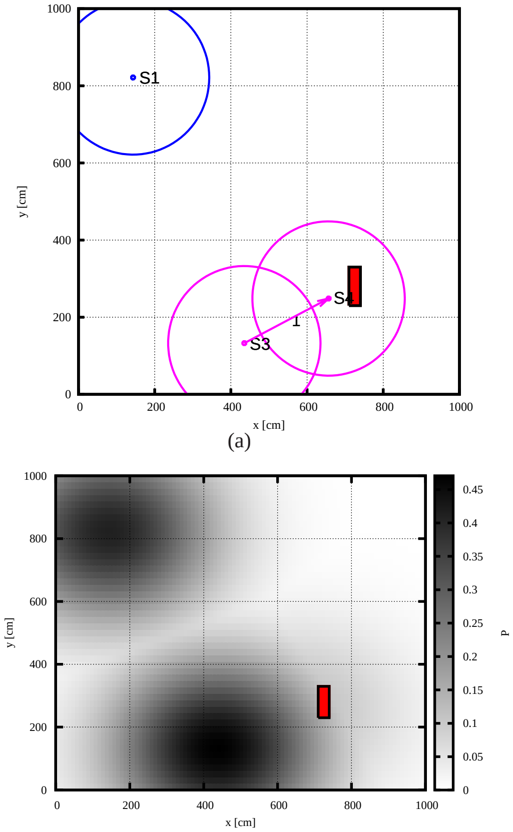
Learning up to Final time instant 13:



(a)

(b)

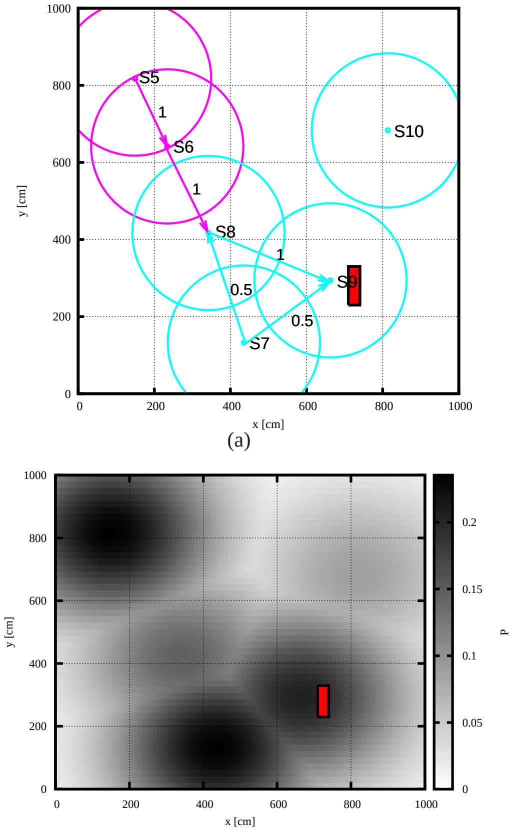
Figure 13: Final hierarchy associated to the position learning context, at instant 13. Figure (a) shows the position of the terminal states and the events. Figure (b) depicts the maximal marginal probability of a state of the hierarchy.



(a)

(b)

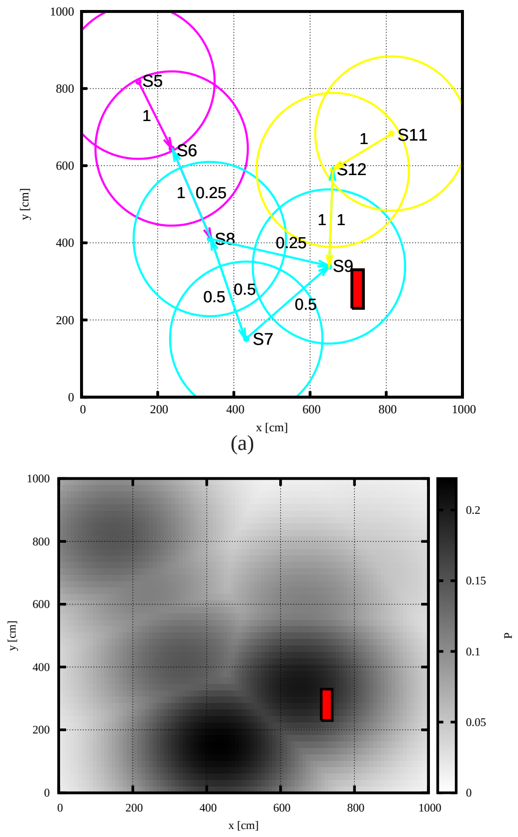
Figure 10: Graphical representation up to instant 3. Figure (a) now also shows the events occurring between the states (arrows with a transition probability). States in blue and magenta represent the first and second level in the hierarchy, respectively. Figure (b) depicts the maximal marginal probability of a state.



(a)

(b)

Figure 11: Graphical representation up to instant 5. Figure (a) shows the position of the terminal states and the events. Cyan colour a state on the third level. Figure (b) depicts the maximal marginal probability of a state.



(a)

(b)

Figure 12: Graphical representation up to instant 7. Figure (a) shows the position of the terminal states and the events. Yellow colour a state on the fourth level. Figure (b) depicts the maximal marginal probability of a state.

The final result for the hierarchy of this illustration example is depicted in Figure 14. This figure shows that the hierarchy has arrived to a stable state since time instant 7. In Figure 13 only slight differences can be observed, with some few new events and slight modifications in the probability map.

This illustration has served to show the incremental nature of the proposed event learning approach. The hierarchy of states and events has shown a consistent behaviour on representing the frequency of states and events induced by the persons of the illustration example.

5.2. *Exploiting the Hierarchy and the Effect of Acuity*

The hierarchy learnt by MILES concentrates rich information, which can vary according to the attributes selected for the learning process. Figure 15 shows three different types of information extracted from the hierarchy, for an application to study the behaviour of a rat, consisting in 4850 frames. The utilised learning context considers three numerical attributes: 2D position attributes **X** and **Y**, and also 2D velocity magnitude attribute **V2D**. A video showing the evolution of the incremental learning process is available[2]

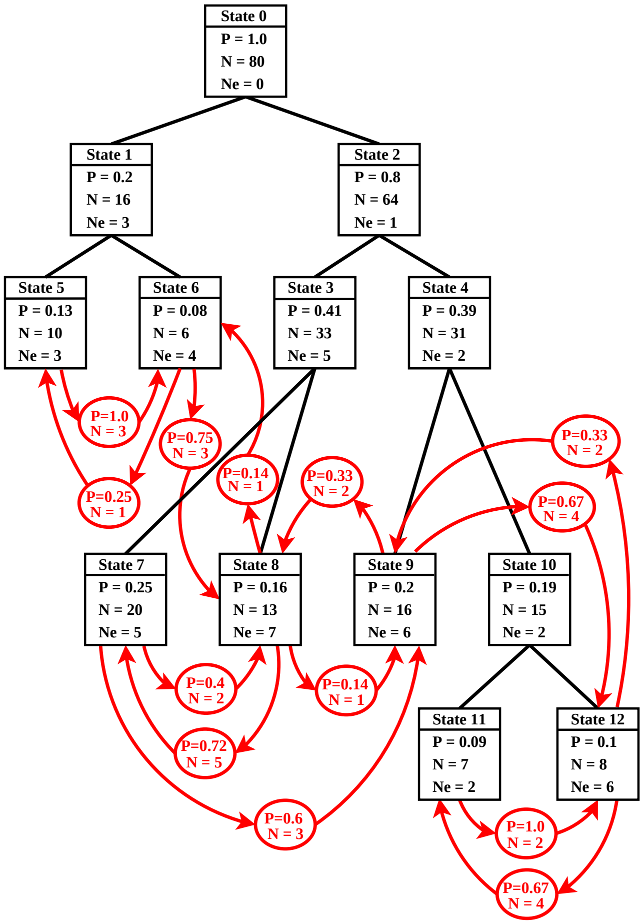
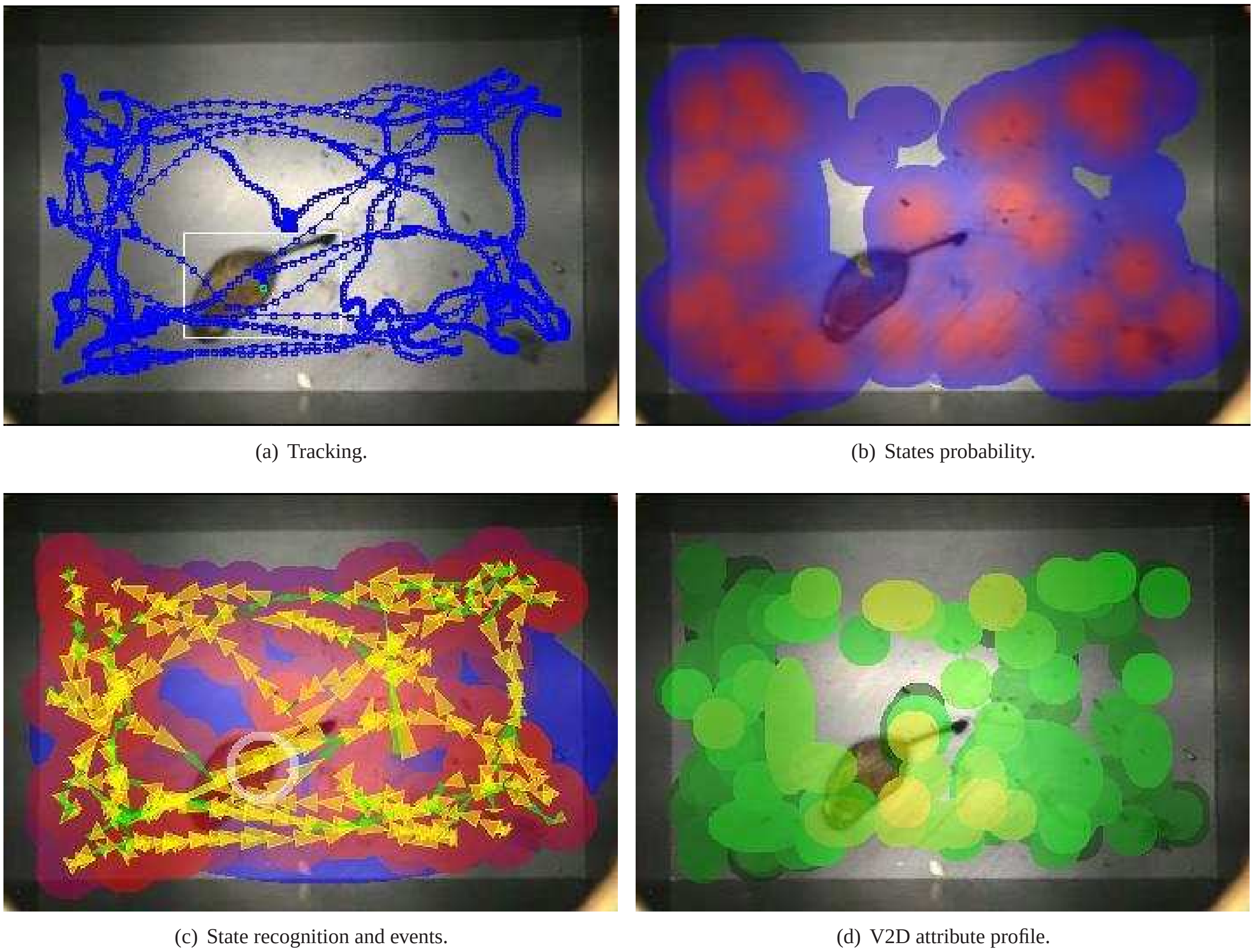


Figure 14: Final hierarchy obtained up to instant 13. For simplicity, only events between terminal states are displayed.

[2] MILES information video available at:
http://profesores.elo.utfsm.cl/~mzuniga/milesX4.mp4


(a) Tracking.

(b) States probability.

(c) State recognition and events.

(d) V2D attribute profile.

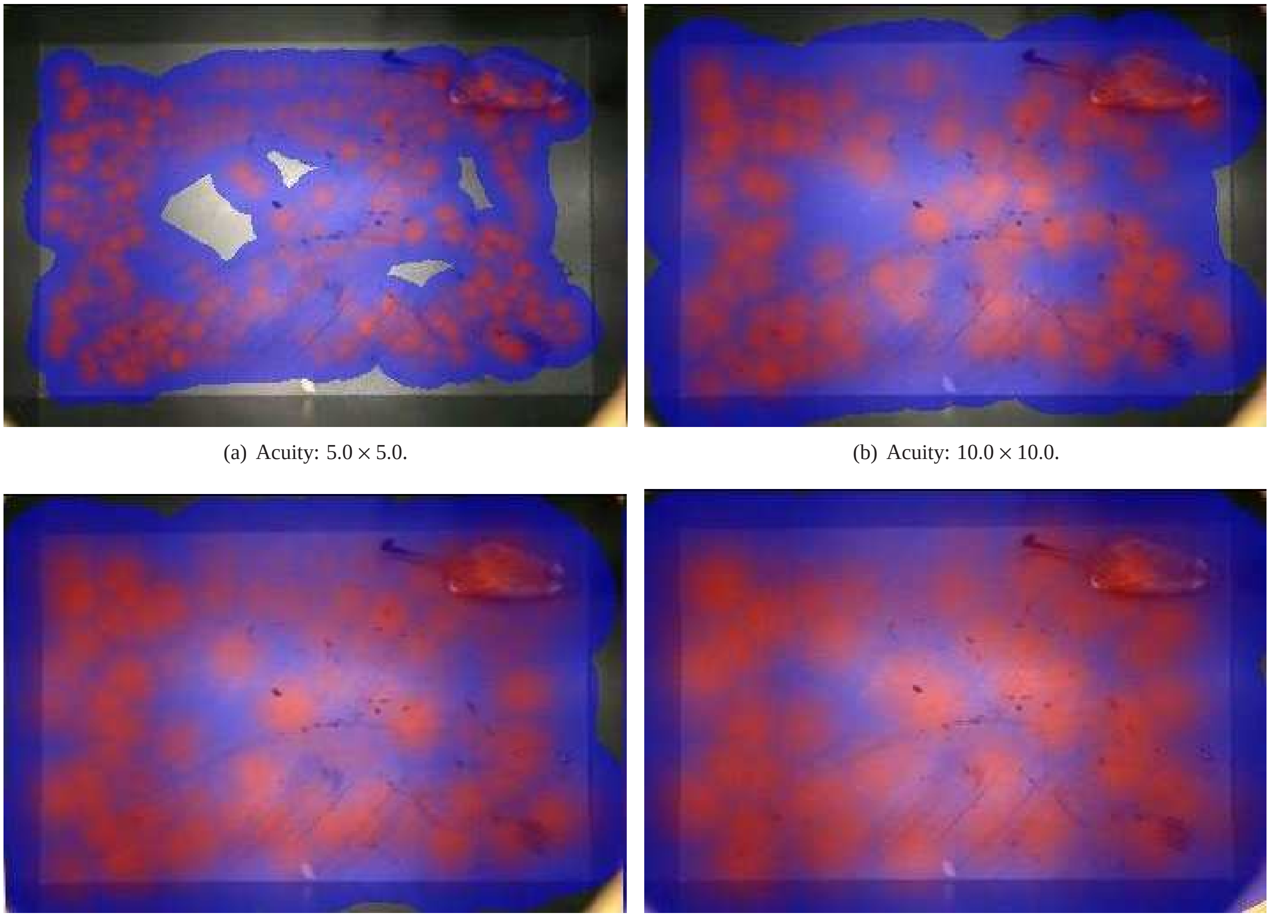
Figure 15: Different information extracted from MILES hierarchy. Image (a) image represents the input from tracking. Image (b) shows the maximal probability for each point, using likely states from the hierarchy (red to blue, for highest to lowest probability). Image (c) shows the same likely states from the hierarchy, only showing their peak probability, and also the events connecting these states. The events are represented with a triangle opening from the starting state to the arriving state (yellow to green, for highest to lowest probability). Recognised states are presented with a white ring. Finally, image (d) shows the behaviour of the **V2D** attribute according to the position (yellow to green, for highest to lowest velocity magnitude). Note that it can be easily inferred that the rat stops at corners and accelerates the most through the widest part of the experimental zone.

We have chosen position and velocity attributes because they can be more easily represented in the input video, but nothing limits the number or nature of the attributes to be learnt. The input information is obtained from a multi-hypothesis tracking approach which is able to compute reliability measures for object attributes, and is described in detail in [20]. It is important to notice that the presence of one or many objects in the video sequence is not relevant for MILES learning process to properly work, as the attributes are learnt each frame from any mobile object which matches with any of the classes defined in the learning context.

There must be certainly many ways of extracting information from the hierarchy. In this particular case, the states are selected searching for the deepest state with a probability higher than a threshold, to obtain relevant states according to the application. There are also many ways to consider the state probability to select the states. For example, we can just consider the probability of the state only, or the conditional probability considering attributes of interest, or even considering these attributes probability weighted by their reliability. In the presented case, we use a conditional probability considering the probabilities of **X** and **Y** attributes, so that likely states with low intra-class similarity are not considered.

The extracted information can then serve, for instance: to determine the more likely (or unlikely) zones according to their probability (figure 15, upper right), which is useful for abnormal behaviour detection and traffic frequency analysis, among many other applications; to determine the likely (or unlikely) behaviours through chains of events (figure 15, lower left), certainly useful for behaviour analysis; and understanding the relations between attributes as, for example, estimating which are the zones where the rat is static or moves quicker (figure 15, lower right).

Other element that has a notorious effect on the results is the **acuity** of each numerical attribute. As previously discussed, the acuity allows the users to define their interest on an attribute. Then, there is no ideal value for this parameter, as it depends on the application. Figure 16 depicts the effect of different val-



(a) Acuity: 5.0×5.0.

(b) Acuity: 10.0×10.0.

(c) Acuity: 15.0×15.0.

(d) Acuity: 20.0×20.0.

Figure 16: Figures show the state probability map results, considering different acuity values (5.0, 10.0, 15.0, and 20.0) for image coordinate attributes **X** and **Y**.

ues of acuity on the probability map. A video showing the incremental evolution of the probability map, for different acuity values, is also available [3]

The figure shows how the state probabilities are affected with lower probability peaks and more plain probability distributions when acuity increases. This is the expected behaviour as, when an user defines a higher acuity, is implicitly saying that higher differences are not significant to the application so that the related instances can be clustered in the same state.

If acuity is increased, also the number of instances similar to a state. Then, the number of states and events is decreased, as shown in Figure 17.

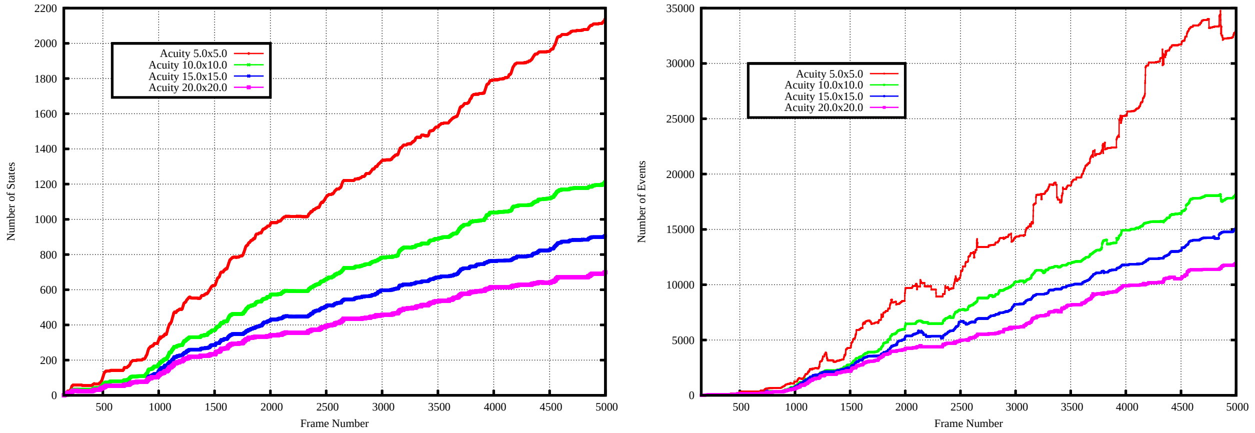
5.3. Symbolic Attributes and Recognition Capabilities

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 18. The videos correspond to an elderly man (Figure 18(a)) and an elderly woman (Figure 18(b)), 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 the same tracking method, previously described, and presented in [22]. A learning context for the class **Person**, combining both numerical and symbolic attributes, was tested considering the following attributes: **3D position** (x, y); symbolic **Posture**, with values for **Standing** or **Crouching** posture; and interaction symbolic attributes $SymD_{table}$, $SymD_{sofa}$, and $SymD_{kitchen}$ between the person and three objects present in the scene (table, sofa, and kitchen table). The possible symbolic values are: 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 with 3D polygons. All the attributes are automatically computed by a tracking method, which is able to calculate the reliability measures of the attributes [22].

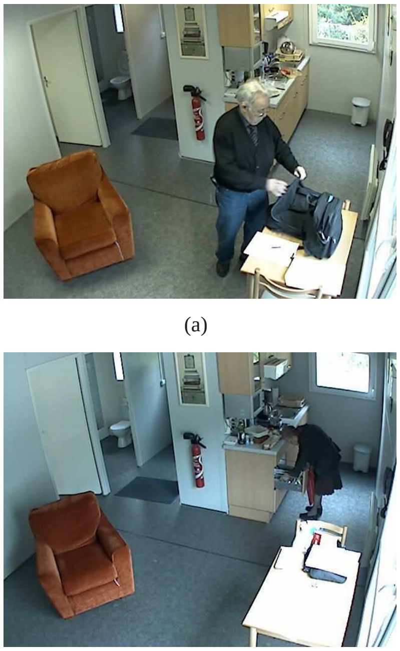
The learning process applied over the 68000 frames have resulted in a hierarchy of 670 state concepts and 28884 event concepts. From the 670 states, 338 state concepts correspond to terminal states (50.4%). From the 28884 events, 1554 event concepts correspond to events occurring between terminal states

[3] MILES acuity video available at:
`http://profesores.elo.utfsm.cl/~mzuniga/acuityX4.mp4`



(a) (b)

Figure 17: Figures (a) and (b) respectively show the number of states and events at each frame, for each acuity value, obtained by processing the rat experiment video.



(a)

(b)

Figure 18: Video sequences for elderly care at home application. Figures (a) and (b) respectively show the observed elderly man and woman.

(5.4%). This evaluation consists in comparing the recognised events with the ground-truth of a sequence, utilising the proposed symbolic-numeric learning context. 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 19.

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 recognised two nonexistent 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.

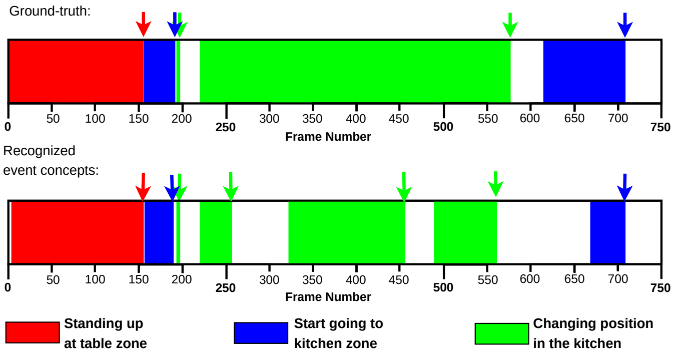


Figure 19: Sequence of recognised events and ground-truth for the elderly woman video. The coloured arrows represent the events, while coloured zones represent the duration of a state before the occurrence of an event.

5.3.1. *Recognised Situations and Symbolic Attributes*

It is also very interesting to check how real situations find their representations in the obtained hierarchies. Here two examples with the previously defined learning context:

- **Going from the kitchen to the table:** This situation consists in the analysed person going from the zone near the kitchen, to the table zone, as depicted with the images shown in Figure 20.

  In the obtained hierarchy the situation is described by the states and events depicted in Figure 21.

  Notice that three states representing each of the displayed images in Figure 20. The probability of occurrence of the first state 25 is $9888/40000 = 0.25$, as the elderly man spends a long time in the kitchen zone. Notice that this state is well describing the fact that the man is all the time

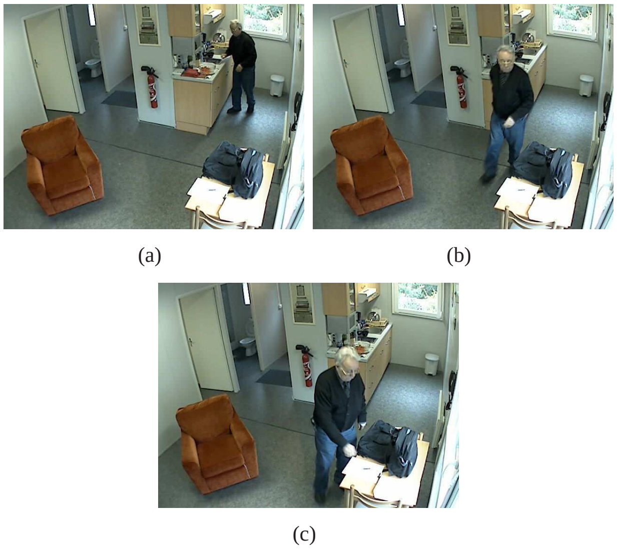
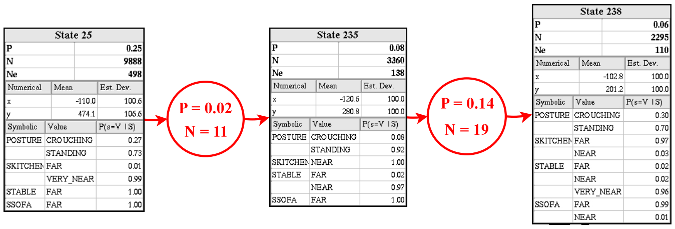

(a)
(b)
(c)

Figure 20: Situation where the person goes from the kitchen to the table. Figures (a), (b), and (c), in this order, describe the way this situation occurs in the scene.



State 25		
P		0.25
N		9888
Ne		498
Numerical	Mean	Est. Dev.
x	-110.0	100.6
y	474.1	106.6
Symbolic	Value	P(s=V |S)
POSTURE	CROUCHING	0.27
	STANDING	0.73
SKITCHEN	FAR	0.01
	VERY_NEAR	0.99
STABLE	FAR	1.00
SSOFA	FAR	1.00

State 235		
P		0.08
N		3360
Ne		138
Numerical	Mean	Est. Dev.
x	-120.6	100.0
y	280.8	100.0
Symbolic	Value	P(s=V |S)
POSTURE	CROUCHING	0.08
	STANDING	0.92
SKITCHEN	NEAR	1.00
STABLE	FAR	0.02
	NEAR	0.97
SSOFA	FAR	1.00

State 238		
P		0.06
N		2295
Ne		110
Numerical	Mean	Est. Dev.
x	-102.8	100.0
y	201.2	100.0
Symbolic	Value	P(s=V |S)
POSTURE	CROUCHING	0.30
	STANDING	0.70
SKITCHEN	FAR	0.97
	NEAR	0.03
STABLE	FAR	0.02
	NEAR	0.02
	VERY_NEAR	0.96
SSOFA	FAR	0.99
	NEAR	0.01

Figure 21: Representation of the situation where the person goes from the kitchen to the table in the hierarchy obtained for the learning context *Position – Posture – SymbolicDistance*.

very near of the kitchen, also showing that at this state the man is not standing all the time, but also crouching approximately a quarter of the total of time spent at this state.

For the same reason that the elderly man spends a long time in the kitchen zone, the events generated for this state are concentrated between states occurring in the kitchen and the conditional probability of the first event is very low (0.02). The second state represents an intermediate passage zone near the kitchen and the table, where the person passes most of the time standing. The third state represents the position very near the table. Here, the person has a crouching posture approximately a third of the total time spent in this state.

- **Crouching and then standing at the table:** This situation consists in the analysed person passing to a crouching posture and then returning to the standing posture, at the zone near the table, as depicted with the images shown in Figure 22.

  In the hierarchy obtained from the previously defined learning context, the situation is described by the states and events depicted in Figure 23. Notice that three states representing each of the displayed images in Figure 22. The probability of occurrence of the first state 131 is not very high 0.04, as the elderly man does not spend a long time in the table zone, compared with the time spent in the kitchen zone. This state is describing that the man is all the time very near of the table at a standing posture. The first event has a high conditional probability (0.4). The second state represents a person still very near of the table but now in a crouching posture.

  Notice also that the utilisation of **posture** and **proximity-to-object** symbolic attributes help the user to bridge the semantic gap of the representation, when needed. Nevertheless, the high number of event transitions between these states, compared with the observed video, highlights a problem inherent to the discretisation process to obtain symbolic attributes: the error is amplified. Here the situation can be that the person, because of errors in the estimation of the dimensions (due to a bad segmentation), gave as result the wrong posture, forcing wrong transitions between both states.

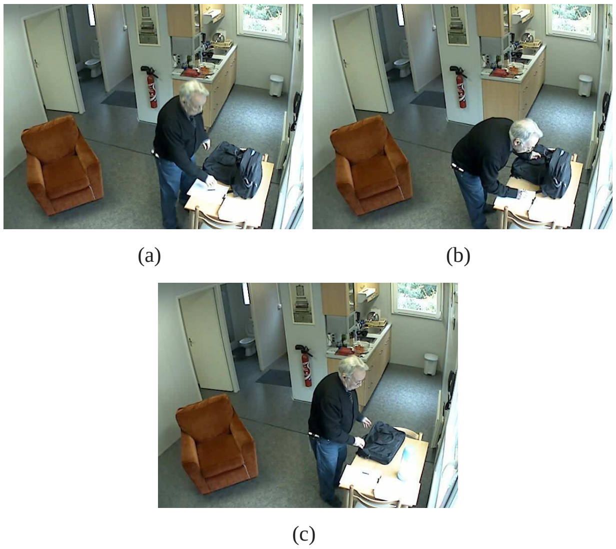
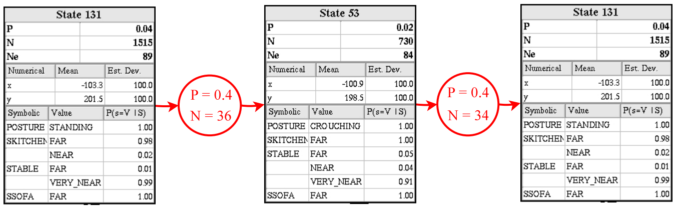

(a)
(b)
(c)

Figure 22: Situation where the person passes to the crouching posture and then returns to the standing posture, near the table. Figures (a), (b), and (c), in this order, describe the way this situation occurs in the scene.



State 131		
P		0.04
N		1515
Ne		89
Numerical	Mean	Est. Dev.
x	-103.3	100.0
y	201.5	100.0
Symbolic	Value	P(s=V |S)
POSTURE	STANDING	1.00
SKITCHEN	FAR	0.98
	NEAR	0.02
STABLE	FAR	0.01
	VERY_NEAR	0.99
SSOFA	FAR	1.00

State 53		
P		0.02
N		730
Ne		84
Numerical	Mean	Est. Dev.
x	-100.9	100.0
y	198.5	100.0
Symbolic	Value	P(s=V |S)
POSTURE	CROUCHING	1.00
SKITCHEN	FAR	1.00
STABLE	FAR	0.05
	NEAR	0.04
	VERY_NEAR	0.91
SSOFA	FAR	1.00

State 131		
P		0.04
N		1515
Ne		89
Numerical	Mean	Est. Dev.
x	-103.3	100.0
y	201.5	100.0
Symbolic	Value	P(s=V |S)
POSTURE	STANDING	1.00
SKITCHEN	FAR	0.98
	NEAR	0.02
STABLE	FAR	0.01
	VERY_NEAR	0.99
SSOFA	FAR	1.00

Figure 23: Representation of the situation where the person passes to the crouching posture and then returns to the standing posture in the hierarchy obtained for the learning context *Position – Posture – SymbolicDistance*.

5.4. Discussion of the Results

As shown in this evaluation, rich information can be obtained with MILES. The results show that the system is able to learn and recognise meaningful events occurring in the scene and that the hierarchical representation can be very rich in information. Also, the utilisation of symbolic attributes allows an easier semantic interpretation of the states.

The computer time performance of MILES is $1300[frames/second]$ for a video with one tracked object and six attributes, and without considering prior stages in the process (e.g. segmentation and tracking), showing the real-time capability of the learning approach.

6. Conclusion

MILES has shown interesting capabilities for state and event recognition. Results have shown that its incremental nature is useful for real-time 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.

The approach allows to learn a model of the states and events occurring in the scene, when no a priori model is available, also giving to users a high flexibility and control through the utilisation of symbolic attributes, the definition of acuity values and the consideration of reliability measures for controlling the uncertainty of information. It has been conceived for learning state and event concepts in a general way, allowing the definition of simultaneously processed learning contexts. 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.

However, more evaluation is still needed for other type of scenes, for other attribute sets, and for different type of tracked objects. The anomaly detection capability of the approach on a large application must also be evaluated. Future work will be also focused in the incorporation of attributes related to interactions between tracked objects (e.g. meeting someone), automatic verification of stability on state instances before learning, and a general state permanence time model.

[1] J. Carbonell, editor. *MACHINE LEARNING. Paradigms and Methods*. MIT/Elsevier, 1990.

[2] M. Chan, A. Hoogs, R. Bhotika, A. Perera, J. Schmiederer, and G. Doretto. Joint recognition of complex events and track matching. In *IEEE Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR06), Volume II*, pages 1615–1622, New York, NY, 17-22 June 2006.

[3] D. H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2(2):139–172, 1987.

[4] J. Gennari, P. Langley, and D. Fisher. Models of incremental concept formation. In J. Carbonell, editor, *Machine Learning: Paradigms and Methods*, pages 11 – 61, Cambridge, MA, 1990. MIT Press.

[5] M. Gluck and J. Corter. Information, uncertainty, and the utility of categories. In E. L., editor, *Proceedings of the 7th Annual Conference of the Cognitive Science Society*, pages 283–287, New York, 1985. Academic Press.

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

[7] R. Howarth and H. Buxton. Conceptual descriptions from monitoring and watching image sequences. *Image and Vision Computing*, 18(2):105–135, January 2000.

[8] F. Jiang, Y. Wu, and A. Katsaggelos. Abnormal event detection from surveillance video by dynamic hierarchical clustering. In *Proceedings of the International Conference on Image Processing (ICIP07)*, volume 5, pages 145–148, San Antonio, TX, September 2007.

[9] K. McKusick and K. Thompson. Cobweb/3: A portable implementation. Technical report, Technical Report Number FIA-90-6-18-2, NASA Ames Research Center, Moffett Field, CA, September 1990.

[10] F. Nater, H. Grabner, and L. Van Gool. Exploiting simple hierarchies for unsupervised human behavior analysis. In *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2014 –2021, june 2010.

[11] F. Nater, H. Grabner, and L. Van Gool. Visual abnormal event detection for prologed independent living. In *Proceedings of the IEEE Healthcom Workshop on mHealth*, 2010.

[12] C. Piciarelli, G. Foresti, and L. Snidaro. Trajectory clustering and its applications for video surveillance. In *Proceedings of the IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS 2005)*, pages 40–45, Los Alamitos, CA, 2005. IEEE Computer Society Press.

[13] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[14] M. Sridhar, A. Cohn, and D. Hogg. Learning functional object-categories from a relational spatio-temporal representation. In *Proceedings of the 18^{th} European Conference on Artificial Intelligence (ECAI08)*, pages 606–610, Patras, Greece, 21-25 July 2008.

[15] N. Thome and S. Miguet. A hhmm-based approach for robust fall detection. In *9th International Conference on Control, Automation, Robotics and Vision (ICARCV '06).*, pages 1–8, december 2006.

[16] A. Toshev, F. Brémond, and M. Thonnat. Unsupervised learning of scenario models in the context of video surveillance. In *Proceedings of the IEEE International Conference on Computer Vision Systems (ICCV 2006)*, page 10, January 2006.

[17] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding (CVIU)*, 115:224–241, February 2011.

[18] T. Xiang and S. Gong. Video behavior profiling for anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):893–908, May 2008.

[19] N. Zouba, F. Bremond, M. Thonnat, and V. T. Vu. Multi-sensors analysis for everyday elderly activity monitoring. In *Proceedings of the 4th International Conference SETIT'07: Sciences of Electronic, Technologies of Information and Telecommunications*, Tunis, Tunisia, March 2007.

[20] M. Zuniga, F. Bremond, and M. Thonnat. Real-time reliability measure driven multi-hypothesis tracking using 2d and 3d features. *EURASIP Journal on Advances in Signal Processing*, 2011(1):142, 2011.

[21] M. Zúñiga, F. Brémond, and M.Thonnat. Incremental video event learning. In J. H. P. Mario Fritz, Bernt Schiele, editor, *Proceedings of the International Conference on Computer Vision Systems (ICVS2009)*, volume 5815 of *Lecture Notes in Computer Science (LNCS) on Computer Vision Systems*, pages 403–414, Liège, Belgium, 13-15 October 2009. Springer.

[22] M. Zúñiga, F. Brémond, and M.Thonnat. Uncertainty control for reliable video understanding on complex environments. In W. Lin, editor, *Video Surveillance*, chapter 21, pages 383–408. INTECH, February 2011.

Marcos Zúñiga was born in 1978 at Valparaíso, Chile. He is an Assistant Professor of the Electronics Department at Universidad Técnica Federico Santa María (UTFSM), Chile. He obtained his Master degree in Computer Science from UTFSM in 2004. He obtained his PhD degree in Computer Science from INRIA - Nice Sophia Antipolis University, France, in 2008. He currently lectures Computer Vision courses at UTFSM, and conducts research work in multi-target tracking, event learning, projective geometry for 3D modelling and interactions, and medical imaging.

François Brémond is a Research Director at INRIA Sophia Antipolis. He has been the head of the PULSAR team since September 2009. He obtained his Master degree in 1992 from ENS Lyon. He has conducted research works in video understanding since 1993 both at Sophia-Antipolis and at USC (University of Southern California), LA. In 1997 he obtained his PhD degree from INRIA in video understanding and François Brémond pursued his research work as a post doctorate at USC on the interpretation of videos taken from UAV (Unmanned Airborne Vehicle) in DARPA project VSAM (Visual Surveillance and Activity Monitoring). In 2007 he obtained his HDR degree (Habilitation à Diriger des Recherches) from Nice University on Scene Understanding: perception, multi-sensor fusion, spatio-temporal reasoning and activity recognition.

Monique Thonnat is french and was born in 1957. She received in 1980 a diploma of engineer ENSPM and a DEA (Master thesis) in Signal and Spatio Temporal Systems from University of Marseille. In 1982 she received her PhD degree in Optics and Signal Processing from University of Marseille III. Her PhD was prepared in the Spatial Astronomical Laboratory of CNRS. (Subject : interactive data reduction methods for astronomical plates: background restitution and radial velocity computing). She obtained her HDR from Nice Sophia Antipolis university in October 2003 (Subject: Towards Cognitive Vision: Knowledge and Reasoning for Image Analysis and Understanding).