



HAL
open science

An unsupervised learning method for human activity recognition based on a temporal qualitative model

Franck Vandewiele, Cina Motamed

► **To cite this version:**

Franck Vandewiele, Cina Motamed. An unsupervised learning method for human activity recognition based on a temporal qualitative model. International Workshop on Behaviour Analysis and Video Understanding (ICVS 2011), Sep 2011, Sophia Antipolis, France. pp.9. inria-00624367

HAL Id: inria-00624367

<https://inria.hal.science/inria-00624367v1>

Submitted on 16 Sep 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An unsupervised learning method for human activity recognition based on a temporal qualitative model

Franck Vandewiele and Cina Motamed

Laboratoire LISIC, Université du Littoral Côte d’Opale, Calais, France
{vandewiele, motamed}@lisic.univ-littoral.fr

Abstract. In this paper, we investigate the problem of monitoring human activities using a network of sensors, including video cameras, in a smart home environment. We introduce an unsupervised method for mining a new kind of qualitative temporally structured activity models from sensor data. We present an application of our method to the recognition of activities of daily living in an elderly care context.

1 Introduction

Smart home projects and pervasive applications generally rely on ubiquitous dedicated sensors for light, contact, motion, temperature, etc. These classical smart home sensors can be used as the building blocks of behaviour detection and recognition algorithms [1]. Because they raise privacy issues, video sensors are generally not used in such environments. Nevertheless, the rich information provided by cameras can offer additional redundancy in case of sensor malfunction and can be very useful in situations where smart home sensors alone do not perform very well, like when several inhabitants perform multiple tasks at the same time. Video cognitive systems themselves can benefit from the adjunction of non-video sensors [2]. Of the few smart home projects using both vision and non-vision sensors, most adopt a supervised learning approach which does not transfer very well from one home to another [3].

We will present an unsupervised approach for behaviour discovery and recognition based on a network of cameras and smart home sensors.

2 Related work

Tapia et al. [4] introduced a system to recognize activities in a sensor-equipped home. This method, based on a systematic and manual annotation of their activities by the inhabitants over several weeks, has its limits: heavily depending on the inhabitants’ good will, the learning process might fail due to inaccurate descriptions, not to mention the weariness caused by the repetitive annotation tasks.

Unsupervised techniques can be applied to human behaviour learning : in [5], video scenes are automatically partitioned into functional categories. Algorithms to discover and recognize frequent trajectories in video scenes are presented in [6].

Several smart home projects have inspired data mining learning methods for human behaviour. In [7], a transactional database is generated from the stream of sensors states. intertransaction mining provides an estimate of temporal relations between frequent itemsets. However, converting to itemsets implies a discretization of the sensor output and might as well split natural activities apart.

Several works [8,9] use episode mining techniques to discover patterns directly from the event stream. Based on a sliding window scanning, these approaches identify either parallel episodes, completely lacking in structure, or serial episodes, heavily structured. Alone, these rather crude models do not reflect the very nature of human activity, which is both repeating and variable. More complex, generic episodes are built around a partial order between symbols. They tend to represent some kind of structured variability. Recently, an algorithm to mine closed strict episodes was proposed in [10]. However, this method provides a way to identify patterns that actually *occur* in the data, and does not grasp the unpredictable nature of human behaviour very well.

3 Low-level event detection

In our setup, tracking of the inhabitants with visual sensors provides a position information. To get a robust positioning, we fuse the information from several cameras. The plane is partitioned into a grid of regions and positioning is converted into a region symbol. Region symbols are merged with low-level information from on/off sensors, resulting in a stream of events emanating from both video and non-video sensors. We then use episode mining techniques on the resulting timestamped sequence of symbols to extract behaviour features.

4 Episode-based model mining

4.1 Definitions and notations

Given a finite alphabet of symbols A , a *sequence* of timestamped symbols of *size* n and *width* $T_e - T_s$ is a triple (s, T_s, T_e) , where $s = ((a_1, t_1), (a_2, t_2), \dots, (a_n, t_n))$, with, for all i : $a_i \in A$, $T_s \leq t_i < T_e$ and $t_i \leq t_{i+1}$ for all i . T_s and T_e are the starting and ending times of s . The *window* of s *starting* at t_s and *ending* at t_e is the sequence (w, t_s, t_e) , where w is the tuple made of those pairs (a_i, t_i) of s where $t_s \leq t_i < t_e$. A *subsequence* of s of *size* m ($m \leq n$) is a sequence $(s', t_{\sigma(1)}, t_{\sigma(m)})$ where $\sigma : \{1, \dots, m\} \rightarrow \{1, \dots, n\}$ is an injective strictly increasing mapping and $s' = ((a_{\sigma(i)}, t_{\sigma(i)})_{1 \leq i \leq m}$. $\mathcal{W}(s, w)$ denotes the set of windows of s of width w and $\mathcal{S}(s, m)$ the set of subsequences of s of size m .

An *episode* ε is a triple (V, \leq, g) , where V is a set of nodes, \leq is a partial order over V and g is a mapping from V to A . Informally, an episode describes a partial order over a multiset of symbols. An episode ε can be represented by a directed acyclic graph G , where $G = (V, E)$, $E \subset V \times V$ and $(v_1, v_2) \in E \Leftrightarrow v_1 \leq v_2$. ε is a *serial episode* if \leq is a total order over V ; ε is a *parallel episode* if \leq is trivial, and we often write $\varepsilon = (V, g)$ for short. $\varepsilon' = (V', \leq', g')$ is a *subepisode* of ε if there exists an injective map $f : V' \rightarrow V$ such that $g'(v) = g(f(v))$ for all $v \in V'$ and f preserves the order relations from V' to V . Two episodes $\varepsilon = (V, \leq, g)$ and $\varepsilon' = (V', \leq', g')$ *have the same nodes* if there exists a bijective mapping $f : V \rightarrow V'$ such that $g(v) = g'(f(v))$ for all $v \in V$. A parallel episode is always the subepisode of an episode having the same nodes. In this sense, parallel episodes are a very generic class of episodes. Serial episodes, on the other hand, are very specific instances of episodes: an episode is always the subepisode of a serial episode having the same nodes.

An episode ε *occurs* in a sequence $s = ((a_i, t_i))_{1 \leq i \leq n, T_s, T_e}$ if there exists an injective mapping $h : V \rightarrow \{1, \dots, n\}$ such that $g(x) = a_{h(x)}$ for all $x \in V$ and for all $x, y \in V$, $(x \leq y \text{ and } x \neq y) \Rightarrow t_{h(x)} < t_{h(y)}$. An episode occurs *frequently* in a sequence s if it occurs in a number of windows of s of fixed width bigger than a given threshold value.

4.2 Statistical structured learning

We assert that a frequent parallel episode ε in a stream of sensor events s is the expression of a frequent behaviour or activity, each instance of this behaviour being represented by an episode having the same nodes as ε occurring in a certain window of s . ε , being unordered, does not say much of the way the activity is usually undertaken. On the other hand, favoring any other episode having the same nodes, be it frequent, as an expression of the behaviour may be too much of a constraint, considering the inherent variability of human activity.

For these reasons, given a frequent episode, we do not model the corresponding behaviour with an episode, but rather with a statistical representation of the episodes having the same nodes and occurring in s . We will use a multigraph with edges labelled by the number of occurrences of a defined order between symbols in all the occurrences of ε .

More formally, given a frequent parallel episode $\varepsilon = (V, g)$ in windows of width w of the sequence s , we define the statistical structure of ε as the edge-labeled directed multigraph (V, E, g, l) , with edges $E = \{(v_i, v_j) \in V \times V, i \neq j\}$, and l an edge-labelling mapping from E to \mathbb{N} , with, for all $(v_i, v_j) \in E$:

$$l(v_i, v_j) = \sum_{W \in \mathcal{W}(s, w)} \left| \left\{ p \in \mathcal{S}(w, |V|), \exists \varepsilon' = (V, \leq, g), \right. \right. \\ \left. \left. \varepsilon' \text{ is serial, } \varepsilon' \text{ occurs in } p \text{ and } v_i \leq v_j \right\} \right| \quad (1)$$

To discover statistical structures from a sequence of events, we first discover frequent parallel episodes using the WINEPI [11] algorithm.

The width of the window used to scan the training data is a critical parameter which has a heavy influence on the quality of the results. Moreover, acceptable choices for this parameter are not transferable from one dataset to an other.

During a second pass, whenever a frequent parallel episode enters the sliding window, serial episodes having the same nodes are discovered and used to update the models.

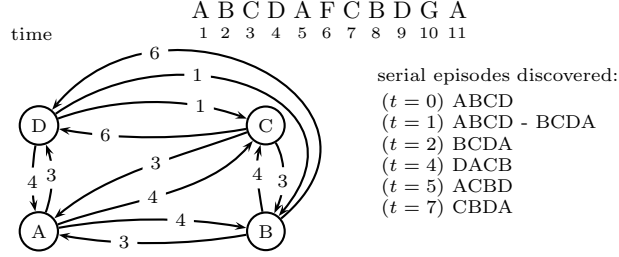


Fig. 1. Example of the statistical structure of the episode ABCD generated with a sliding window containing 5 symbols. A list of the serial episodes discovered is given, with the starting times of the windows the episodes occur in. For instance, the label on the edge from A to B means that the relation $A \leq B$ is true in 4 of the occurrences of the episode ABCD.

4.3 Temporal qualitative model

The statistical structure (V, E, g, l) of a frequent episode ε can be used to describe how often two symbols occur in a defined order in the occurrences of ε . Given two symbols $a = g(v_i)$ and $b = g(v_j)$, the probability $\mathbb{P}_\varepsilon(a \leq b)$ that $a \leq b$ is true in an occurrence of ε is defined by:

$$\mathbb{P}_\varepsilon(a \leq b) = \mathbb{P}_\varepsilon(v_i \leq v_j) = \frac{l(v_i, v_j)}{l(v_i, v_j) + l(v_j, v_i)} \quad (2)$$

We build the temporal qualitative model $\bar{\varepsilon}$ of a frequent episode ε by discarding those edges of the associated model whose probabilities are lower than a given threshold k . That is: $\bar{\varepsilon} = (V, E, g)$ where $(v_i, v_j) \in E \Leftrightarrow \mathbb{P}_\varepsilon(v_i \leq v_j) \geq k$. Note, as depicted in figure 2, that $\bar{\varepsilon}$ may not be an acyclic graph and therefore not the representation of an episode.

5 Score-based recognition system

During the recognition process, a window is sliding on the stream of time-stamped symbols to detect incoming occurrences of frequent parallel episodes.

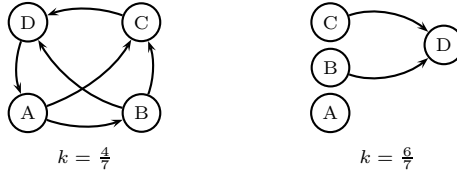


Fig. 2. Qualitative models of the episode ABCD from figure 1 with $k = \frac{4}{7}$ and $k = \frac{6}{7}$.

Each occurrence is associated to a qualitative score reflecting whether it is coherent with the associated qualitative model. For each pair of symbols, the score of the subsequence is increased by 1 if these symbols appear in the same order in the model, decreased by 1 if they appear in the opposite order and remains unchanged if these symbols are unordered in the model.

Formally, let $\bar{\varepsilon} = (V, E, G)$ the qualitative model of a frequent episode ε and (s, T_e, T_s) a minimal subsequence covering ε , with $s = (a_i)_{i \in I}$. The score of s with respect to $\bar{\varepsilon}$ is $\mathcal{S}(s) = \sum_{i < j} \sigma_{i,j}$, where, if $a_i = g(u)$ and $a_j = g(v)$, $\sigma_{i,j}$ is defined as:

$$\sigma_{i,j} = \begin{cases} 1 & \text{if } (u, v) \in E \\ -1 & \text{if } (v, u) \in E \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

6 Experimentation

We carried an experimentation using the data collected at the Ger’home laboratory [12, 13]. The laboratory is equipped with 4 video cameras, and a set of on/off sensors. The dataset consists of 11 sequences of 4 hours of video and sensor outputs. During each experimentation, volunteers were asked to perform various activities of daily living, such as preparing and taking meals. For a first experimentation, we ran our learning algorithm on one month time of simulated data and confronted the learned models to the 11 sequences of real data.

6.1 Simulation

Our simulation protocol relies on the following concepts:

- *actors* behaviours are simulated in the *context* of a smart home;
- a context consists of several *zones* (e.g. `kitchen`, `living room`, etc.);
- zones are organized in a *topology* describing which zones are adjacent and how zones are nested into each other (e.g. `kitchen` adjacent to `corridor`, `living room` contains `close_to_table`, etc.);
- actors move from zone to zone;
- zones contain *interactive devices* (e.g. `drawer_upper`, `television`, etc.);
- actors and interactive devices have several *states* (e.g. `position`, `open`, etc.);
- states can have one or more *values* (e.g. `in_kitchen`, `true`, `false`, etc.);
- actors can *interact* with devices, resulting in a change of states values.

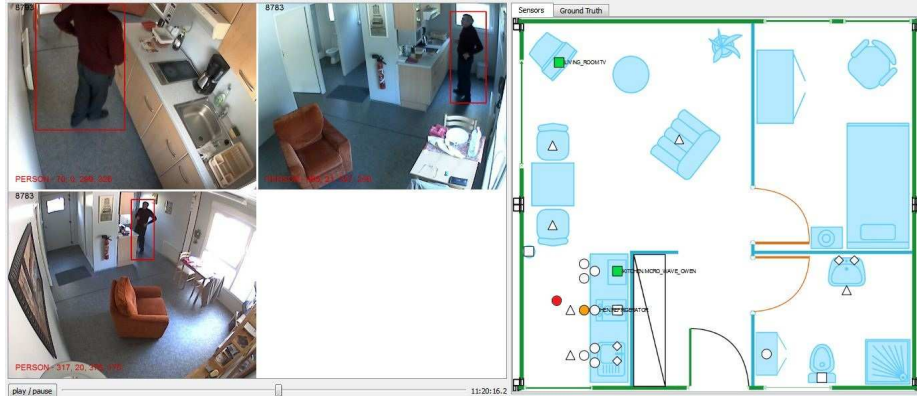


Fig. 3. The Ger'home laboratory setup

- states values are reported through (possibly defective) *sensors*.

Actors behaviours are modeled as follows:

- for each actor, a *schedule* is defined;
- schedules determine when *scenarios* occur (e.g. `prepare_meal` occurs at 11:32, etc.);
- scenarios consist in a succession of *events*: movements (e.g. “go to bathroom”), pauses or interactions with devices (e.g. “switch `light_living_room` on”);
- inside a scenario, events do not always happen in the same order; depending on the occurrences, some events may not as well happen at all;
- events have a *duration* which describes how long an action takes to be completed; durations can follow some random distributions described in the scenario description;
- events can pre-trigger or post-trigger the inclusion of additional scenarios in the schedule (e.g. “switch `light_living_room` on” will trigger “go to living_room” first if the actor is not in the correct zone);
- scenarios are organized in a *hierarchy* (e.g. `toilet` will interrupt `watch_tv` but `take_meal` will not start until `prepare_meal` is completed).

Graphical representations of some simulated data and some real data are depicted in figure 4.

6.2 Performance evaluation

Due to the great variety of the activities performed in the sequences, we focused our attention on the recognition of the activity “preparing meal”. Our first results show that our system is able to learn fragments of activities rather than whole activities.

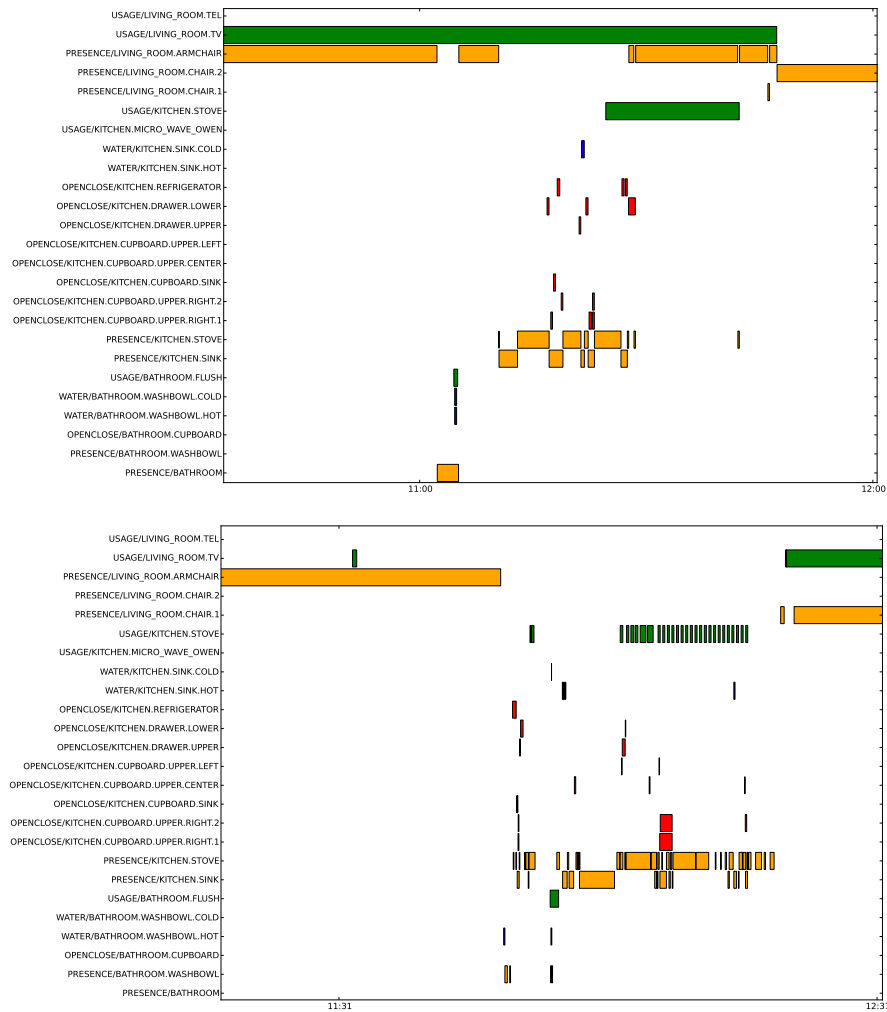


Fig. 4. Graphical representations of some simulated data (top) and some real data from the April, 16 dataset (bottom). The portions of the data shown correspond to the preparation of a meal.

Evaluating the performances of an unsupervised learning process is a difficult task. To measure the performance of our system, we first have extracted 13 episodes learned from the training data occurring when meals are prepared. Next, we tested our recognition system on the 11 sequences of real data and 77% of these patterns were detected.

7 Conclusion

We have proposed a new qualitative model for human activity recognition based on an unsupervised learning approach using episode mining techniques. During the learning step, we first extract frequent episodes from the training data. We introduce a qualitative temporal model describing some of the temporal relations intrinsic to the frequent patterns. Finally, during the recognition process, a qualitative score is proposed to recognize patterns compatible with such models.

Our first experimentation did not use much of the rich information provided by visual sensors. We intend to use more visual features in the future.

The whole process is heavily depending on the choice of a width for the sliding windows and the minimum support for frequent episodes. In future works, we will investigate techniques to automatically adjust these parameters.

Acknowledgment

We thank the pulsar team at INRIA who provided the Ger'home dataset.

References

1. D. J. Cook and S. K. Das, "How smart are our environments? an updated look at the state of the art." *Pervasive and Mobile Computing*, pp. 53–73, 2007.
2. A. Dore, M. Pinasco, and C. Regazzoni, "Multi-modal data fusion techniques and applications," in *Multi-camera networks: Concepts and Applications*, H. Aghajan and A. Cavallaro, Eds. Elsevier, May 2009, pp. 213–237.
3. N. Zouba, F. Brémond, and M. Thonnat, "Multisensor fusion for monitoring elderly activities at home," in *IEEE Int. Conf. on Advanced Video and Signal based Surveillance (AVSS 2009)*, Genoa, Italy, Sept 2009.
4. E. M. Tapia, S. S. Intille, and K. Larson, "Activity recognition in the home using simple and ubiquitous sensors." in *Pervasive'04*, 2004, pp. 158–175.
5. M. W. Turek, A. Hoogs, and R. Collins, "Unsupervised learning of functional categories in video scenes." in *ECCV (2)'10*, 2010, pp. 664–677.
6. C.-L. Liu, E. Jou, and C.-H. Lee, "Analysis and prediction of trajectories using bayesian network." in *ICNC'10*, 2010, pp. 3808–3812.
7. S. Lühr, G. A. W. West, and S. Venkatesh, "Recognition of emergent human behaviour in a smart home: A data mining approach." *Pervasive and Mobile Computing*, pp. 95–116, 2007.
8. P. Rashidi and D. J. Cook, "An adaptive sensor mining framework for pervasive computing applications." in *KDD Workshop on Knowledge Discovery from Sensor Data'08*, 2008, pp. 154–174.
9. T. Gu, Z. Wu, X. Tao, H. K. Pung, and J. Lu, "epSICAR: An Emerging Patterns based approach to sequential, interleaved and Concurrent Activity Recognition," in *2009 IEEE International Conference on Pervasive Computing and Communications*, Mar. 2009.
10. N. Tatti and B. Cule, "Mining closed strict episodes," in *Proceedings of the 10th IEEE International Conference on Data Mining (ICDM-2010)*, 2010.

11. H. Mannila, H. Toivonen, and A. I. Verkamo, "Discovery of frequent episodes in event sequences." *Data Min. Knowl. Discov.*, pp. 259–289, 1997.
12. N. Zouba, B. Boulay, F. Brémond, and M. Thonnat, "Monitoring activities of daily living (adls) of elderly based on 3d key human postures." in *ICVW'08*, 2008, pp. 37–50.
13. <http://gerhome.cstb.fr/en>.