

Trajectory based Primitive Events for learning and recognizing Activity

Guido Pusiol, François Bremond, Monique Thonnat

▶ **To cite this version:**

Guido Pusiol, François Bremond, Monique Thonnat. Trajectory based Primitive Events for learning and recognizing Activity. Second IEEE International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS2009), Sep 2009, Kyoto, Japan. inria-00503209

HAL Id: inria-00503209
https://inria.hal.science/inria-00503209

Submitted on 16 Jul 2010

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.

Trajectory based Primitive Events for learning and recognizing Activity

Guido Pusiol gtpusiol@sophia.inria.fr
Francois Bremond fbremond@sophia.inria.fr
Monique Thonnat Monique.Thonnat@sophia.inria.fr
Pulsar - INRIA - Sophia Antipolis

Abstract

This paper proposes a framework to recognize and classify loosely constrained activities with minimal supervision. The framework use basic trajectory information as input and goes up to video interpretation. The work reduces the gap between low-level information and semantic interpretation, building an intermediate layer composed Primitive Events. The proposed representation for primitive events aims at capturing small meaningful motions over the scene with the advantage of been learnt in an unsupervised manner. We propose the modeling of an activity using Primitive Events as the main descriptors. The activity model is built in a semi-supervised way using only real tracking data. Finally we validate the descriptors by recognizing and labeling modeled activities in a home-care application dataset.

1. Introduction

The automatic recognition and classification of daily human activities is a topic that remains open. In the literature the computational approaches assume usually prior knowledge about the activities and the environment. This knowledge is used explicitly to model the activities in a supervised manner. For example in video surveillance domain, the technical and scientific progress requires nowadays human operators to handle large quantities of data. It becomes almost an impossible task to continually monitor these data sources manually. It is of crucial importance to build computer systems capable of analyzing human behavior with a minimal supervision.

Computer-based video applications need several processing levels, from low-level tasks of image processing to higher levels concerning semantic interpretation of the monitored scene. At the moment the reduction of the gap between low-level tasks up to video understanding is still a challenge.

This work addresses these problems by presenting a novel framework that links the basic visual information (i.e. tracked objects) to the recognizing and labeling activities (e.g. Working in the kitchen) by constructing an intermediate layer in a completely unsupervised way.

The intermediate layer is composed of meaningful transitions (i.e small trajectories corresponding to primitive events) between two regions of interest. To automatically model these primitive events first the scene topology is learnt in an unsupervised way. Thus the intermediate layer tries to capture the intention of the individual to perform basic tasks, using only minimal information. Using visual information enables to reduce the complexity of systems that usually use numerous sensors to enrich the observation data. Given global annotated activities for one person (i.e. activity Ground Truth), the meaningful patterns of primitive events are extracted to characterize the activities of interest. This activity ground truth bridges the gap between observation and semantic interpretation. The patterns of primitive events are then used as generic activity descriptions in order to recognize automatically the main activities for another observed person.

These contributions are described in the third section. The process to build the scene topology is presented in the fourth section. The generation of primitive events and the modeling of activities are respectively described in the fifth and sixth sections. The paper concludes with validation experiments on home-care monitoring, and explains how typical activity such as "Working in the kitchen" were automatically recognized.

2. Related Work

The data-mining field can provide adequate solutions to synthesize, analyze and extract information. Because of the advance made in the field of object detection and tracking [8] data-mining techniques can be applied on large video data. These techniques consist in classifying multiple video features (e.g. trajectories) in activity categories associated with meaningful semantic keywords that will allow the retrieval of the video. Usually low level features (i.e., color, texture, shape, and motion) are employed. Recently particular attention has been turned to the object trajectory information over time to understand high level activity. The trajectory based methods to analyze activity can be divided in two groups, supervised and unsupervised.

The typical supervised methods proposed such as [14, 15, 17] can build activity behavior models in a very accurate way. The problem is that they require of big training datasets labeled manually.

The unsupervised methods generally include: Neural Networks based, approaches such as [6, 11, 18], they can represent complex nonlinear relations of trajectory space in a

low-dimensional structure, the networks can be trained sequentially and easily updated with new examples. but they require big amount of training data and the complexity of parametrization usually makes the networks become useless after long periods of time.
Clustering approaches such as Hierarchical Methods [7, 13] allow multi resolution activity modeling by changing the number of clusters, but the clustering quality depends on the decision of when to clusters should be merged or spit. Adaptative Methods [2, 3, 9], the number of clusters adapts to changes over time, making possible on-line modeling without the constraint of maintaining a learning data-set. In these methods is difficult to initialize a new cluster preventing outlier inclusion. [1] [19] use dynamic programing based approaches to classify activity, quite effective methods when time ordering constraints hold.
Hidden Markov Model based approaches such as [12, 16] captures spatio-temporal relations in trajectory paths, allowing high level analysis of an activity, is very suitable for detecting abnormalities. These methods need of prior knowledge and the adaptability in time is poor.
Recently Morris and Trivedi [5], learn topology scene descriptors (POI) and modeled the activities between POIs with HMMs encoding trajectory points, the approach is suitable to detect abnormal activities and has good performance when used in structured scenes. The method requires of time order constraints and the topology is based in the entry and exit scene zones. Hamid et al. [4] merges the scene topology and censorial information, modeling sequences of events (n-grams) to discover and classify activity. The method requires manual specification of the scene.
Most of the methods described above can be used in structured scenarios (i.e. highway, or a person a laboratory), and cannot really infer activity semantics. To solve this problems we propose a method capable of recognizing loosely constraint activities in non structured scenes, and we go up to semantic interpretation with minimal human knowledge.

3. Overview

The proposed approach aims first at learning the main everyday activities of a person observed by video cameras given a coarse labeling of these activities. Second the goal is to recognize automatically these activities while observing another person. The approach is composed of 5 steps. First, people are detected and tracked in the scene and their trajectories are stored in a database, using a classical region based tracking algorithm. Second, the topology of the scene is learnt using the regions (called Slow Regions, SRs) where the person usually stands and stops. This topology is a set of logical regions (e.g. "sink region" corresponds to the zone where people wash the dishes) which are usually present during typical everyday activities.
Third, the transitions between these Slow Regions are computed by cutting the observed person trajectory. These transitions correspond to short unit of motion and can be seen as basic elements constituting more complex activities. These transitions are called Primitive Events (PEs) and are learnt thanks to the SRs.
Fourth, a coarse activity ground truth is manually performed on a reference video corresponding to the first monitored person. Thanks to this ground truth, the associated Primitive Event histograms are globally labeled.
Fifth, using these labeled Primitive Event histograms the activities of the second monitored person can be automatically recognized.

4. Scene Topology

The scene topology is a set of SRs learnt through clustering of meaningful slow points. Other features than slow points could have been selected such as changes of direction, acceleration, changes of shape, but slow points are the most salient features to characterize regions of interest.

4.1. Trajectory Slow points

We use the speed of the object as the measure of velocity. The speed of a trajectory point p_i is estimated by the object spatial distance walked in a fixed window of points, centered at p_i. This way we relax the noise of the trajectory due to the tracker. We use a speed threshold H_{SLOW} to compute the trajectory slow points.
Let T be a trajectory, where $T =< p_1, ..., p_n >$:

$$Speed_{p_i} = \frac{dist(p_{i-w}, p_{i+w})}{2 * w} \quad (1)$$

$$p_i \in SLOW \quad if \quad Speed_{p_i} < H_{SLOW} \quad (2)$$

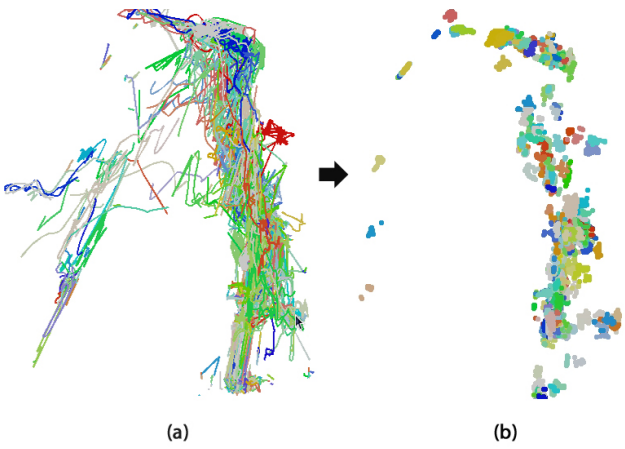


Figure 1. (a) Real-Data Trajectories (DT_{Test}). (b) Extracted SSLPs differentiated by color

4.2. Trajectory strings of slow points

After the computation of trajectory slow points, two situations can appear: isolated slow points or groups of slow

points along the trajectory. Isolated points are usually not meaningful enough to represent logical regions corresponding to an activity (i.e. where an individual interacts with the environment). Thus we perform a secondary calculation by keeping only "strings of slow points" ($SSLP$). We restrict the string size allowing sequences of at least Q_{STAY} points (i.e. the individual maintains the slow motion for 4 seconds or more).
Let S be a sequence $< p_{i-w}, ..., p_{i+w} >$, of a trajectory T, then:

$$S \in SSLP \text{ if } \|S\| > Q_{STAY} \wedge p_{i-w}...p_{i+w} \in SLOW \quad (3)$$

In the figure 1(a) a real-data trajectory data-set of a monitored person living in the experimental apartment, and in 1(b) the subsequence of strings of slow points obtained, differentiated by color. Finally we estimate "scene slow point" (SSL) as the average of points within a sequence. This is motivated by our interest of having a single point representing that the individual stayed in a zone for a certain time.

$$SSL_q = Avg\{SSLP_q\} \quad (4)$$

4.3. Clustering slow points

SSLs points should characterize the region of interest where the individual interacts with static scene objects such as equipment. However SSLs points are not necessarily meaningful. For example the region where a person stops randomly could be an important region that does not represent necessarily any interaction with the environment. To refine the scene regions we perform K-means clustering over the set of SSLs. Here, the selected number of clusters represents the abstraction level of the scene representation. For instance in the figure 2 (a), 6 clusters are extracted, representing interactions with: two sections of the kitchen (shelf and sink), two sections of the table (chairs), the armchair and the exit-hall. Clusters are not always linked with static scene objects, some clusters can be temporarily representing an activity that needs to be considered. For example this is the case when the person moves the chair, uses the chair and moves it back to the original position. For clustering we have tested different types of distance: Euclidean, City-Block, correlation, Kendall's tau, Spearman's rank correlation. The euclidean distance has shown to agglomerate better the SSLPs into meaningful regions.

4.4. Scene Model

The scene topology is modeled by a set of Slow Regions (SRs).
Formally a SR is a triplet of 3 variables.

$$SR_i =< SR_Spatial, SR_Time, SR_Frequency > \quad (5)$$

where:
$SR_Spatial$ is the average of the SSLs in a cluster corresponding to the central point of the cluster.

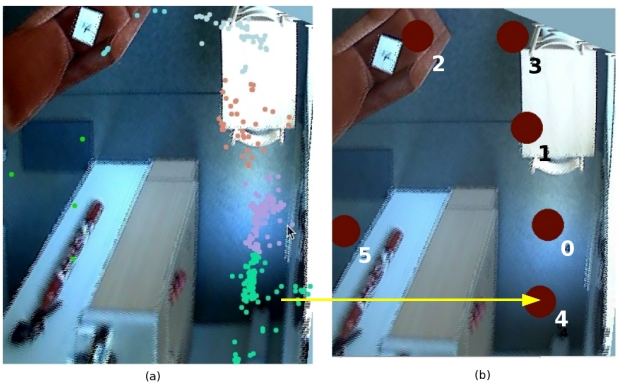


Figure 2. Top View of the apartment kitchen and living room. (a) 6 clusters of SSL points differentiated by color (euclidean-distance), (b) $SR_Spatial$ points numbered from 0 to 5

$SR_Spatial = Avg(\{SSL_t\}) \ \forall \ SSL_t \in Cluster_i$
SR_Time is a Gaussian function describing the time spend by the person at the $SSLs \in Cluster_i$ (extracted from the SSLPs).
$SR_Frequency$ is a Gaussian function that describes the time spent outside the Cluster (i.e. the zone) while revisiting it.
The set of these SRs represents the scene topology. Different granularity of the SRs have been experimented and correspond to different activity abstraction levels. In the following experiments a 8 SRs topology has been used to better represent activities in the kitchen and the table. For notational simplicity a SR is associated to a natural number in the rest of the paper.

5. Primitive Events

For computing Primitive Events we cut a trajectory in to significant segments. The trajectory segments in conjunction with the scene topology information correspond to basic units of motion linked usually with a primitive activity (i.e a person that is in movement stops to do something).

5.1. Cutting trajectories

A trajectory cut is defined as the trajectory segment from the last point of a $SSLP_i$ to the last point of the next $SSLP_{i+1}$ ordered by time of appearance in the trajectory. It is worth nothing that a trajectory cut can pass through a SR without ending there. This is because, an individual crossing a SR without stopping does not necessarily mean that the individual acts or interacts in that region. In particular for our scene in the way from the kitchen to the sofa a person can cross several SRs. An example can be found in the figure 3. Formally given a trajectory $T =< ..., s_1, ..., s_n, p_1, ..., p_m, q_1, ..., q_n, ... >$ where $s_i \in SSLP_a$ and $q_i \in SSLP_{a+1}$, then a trajectory cut TC is:

$$TC =< s_n, p_1, ..., p_m, q_1, ..., q_n > \quad (6)$$

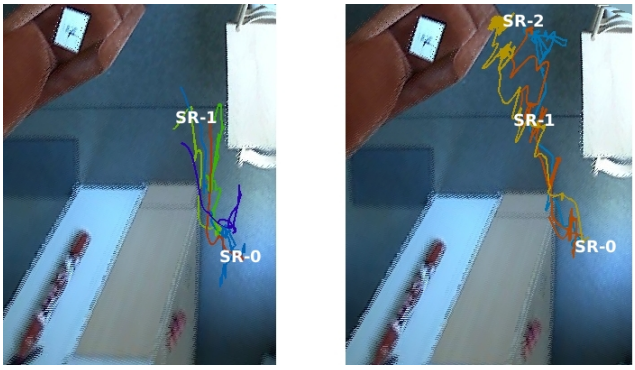


Figure 3. Example of 2 sets of trajectory cuts

5.2. Primitive Event Extraction

The fusion of trajectory cuts and scene Topology information is used to build Primitive Events (PE). A PE is represented as a sixplet.

$$PE =< SSR, ESR, TI, Q, SF, EF > \quad (7)$$

Given a trajectory cut $TC =< p_0, ..., p_n >$ then:
SSR "Start Slow Region" is the label of the nearest SR (Slow Region) of the scene topology to p_0

$$SSR = SR_i \text{ if } dist(p_0, SR_i) < dist(p_0, SR_j) \ \forall j \neq i \quad (8)$$

ESR "End Slow Region" is the label of the nearest SR of the scene topology to p_n.

$$ESR = SR_i \text{ if } dist(p_n, SR_i) < dist(p_n, SR_j) \ \forall j \neq i \quad (9)$$

IM "Imprecision" represents the distance of PE to a perfect motion between SRs.

$$IM = dist(p_0, SSR) + dist(p_n, ESR) \quad (10)$$

T "Time" is the duration of the trajectory cut (normalized by the video Frame Rate).

$$T = \frac{\|TC\|}{Frame_Rate} \quad (11)$$

SF and EF represents the starting and ending frames of TC.
$SF = TC_{startframe}$
$EF = TC_{endframe}$
In this work we do not consider a PE when its imprecision has a too high value.
The primitives events are classified by type depending on their SSR and ESR. Thus for notational simplicity we label them as $(SSR - ESR)$ (i.e (3-4) stands for a PE between SRs with ID 3 and 4).
In the figure 4 an example of 3 extracted primitive events is displayed. These primitive events are used as basic elements to build more complex activities.

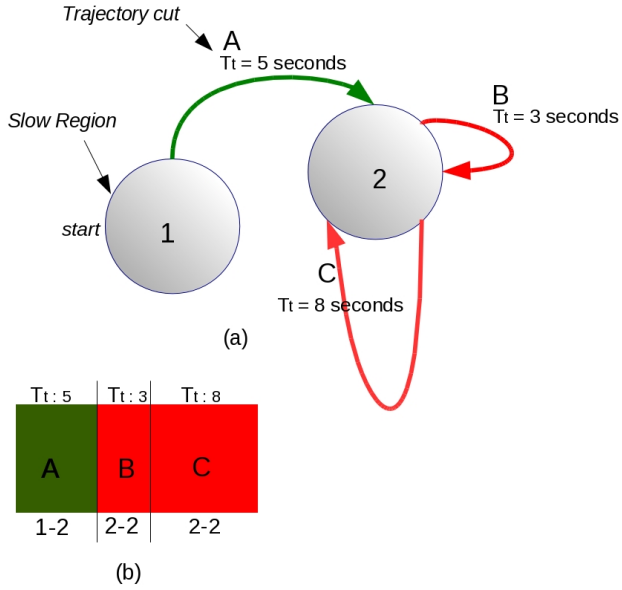


Figure 4. (a) Graphical flow of trajectory cuts A,B and C over the time. (b) graphical representation of the obtained PEs displayed on a time line. From B and C we obtain the same PE label but they are differentiated by the other features (T, SF, EF, IM).

6. Activity Modeling

Instead of manually specifying the possible activities for a scene, which in some cases can become difficult, a model for each activity is automatically learnt through the observation of the subsequence of PEs occurring in a time window. The duration and the label of the activity are specified by the user by building a set of coarse Activity Ground Truth covering the whole video length for one monitored person as illustrated in figure 5 (b).
In figure 5 (a) a graphical representation of the sequence of automatically detected Primitive Events is displayed for the whole length of dataset1 (DT_{Test}). Graphically we differentiate each PE type by color. White segments appear due to one of these three reasons: The tracker has lost the person, The person is not in scene, The PE is filtered because of a poor precision measurement.

6.1. Ground truth

We define the activities of an individual in a video sequence, by building an activity ground truth. We built two ground-truth (GT_{Test} and GT_{Learn}), from 2 datasets[1] of 2 individuals (DT_{Test} and DT_{Learn}) -figure 6-. We use GT_{Learn} for learning the activity models of DT_{Learn} and GT_{Test} for validation of the DT_{Test} detected activities.

6.2. The Model

The features we use to model an activity are the PEs contained in the time window when the activity occurs (fig

[1] http://www-sop.inria.fr/pulsar/personnel/Francois.Bremond/topicsText/gerhomeProject.html

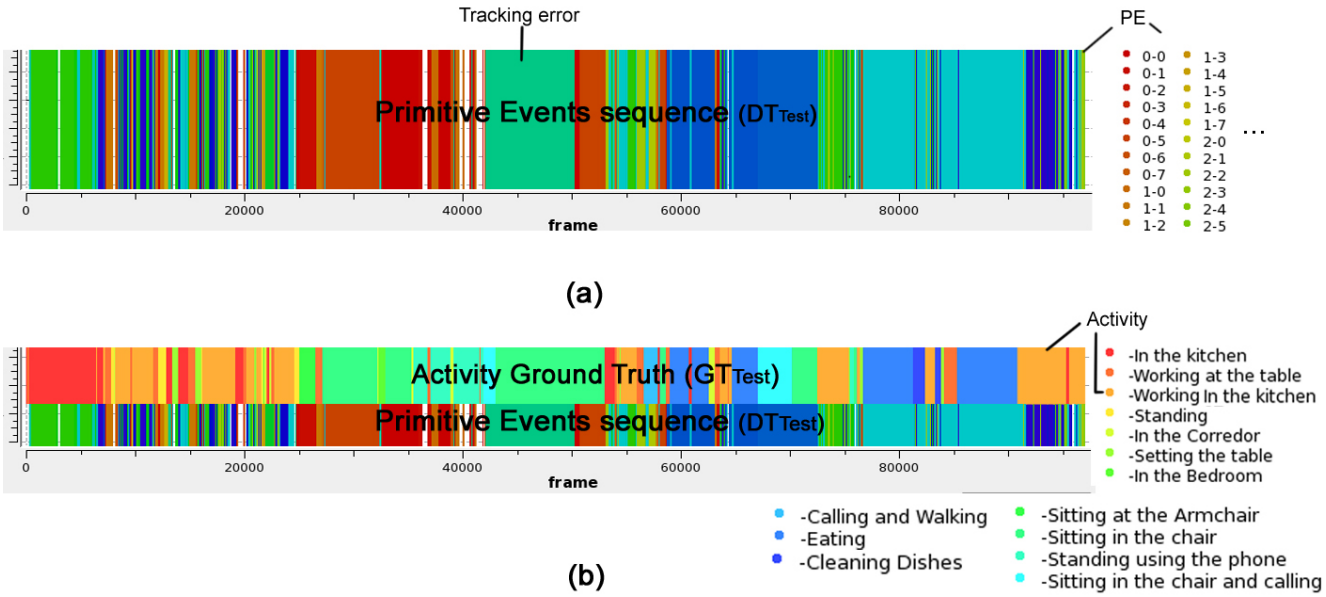


Figure 5. (a) Graphical representation of PE sequence of DT_{Test} in the time-line, the color is the PE type and the with the spend time. (b) The overlapping of GT_{Test} activities, and the PE sequence of DT_{Test}



Figure 6. Snapshots of DT_{Test} and DT_{Learn} videos (4 and 3 hours length respectively).

7). The window is obtained by aligning the detected PEs with the ground truth activities.

Concerning this point modeling the temporal relations by HMMs and n-grams have already been proposed in the literature, but the targeted activities (i.e. homecare monitoring) are loosely constraint and does not have strong structural temporal patterns requiring a specific temporal processing.

We have also evaluated the probability of a PE to belong to an activity using Bayesian rule. The results have suggested that more datasets are required to provide a reliable probabilistic description of the activities.

Thus we have found that a simple histogram containing the instances of the PEs during the activity, captures sufficiently the characteristic PEs to describe and differentiate the activities. Also when two different activities share a similar spatial location, the key feature to differentiate both is the activity length duration. This motivates us to encode the time accumulation of the PEs in a second histogram.

More formally, a first histogram ($H1$) contains the number of instances for each PE type appearing in the window. The second one ($H2$) represents the accumulation of the time spend during the primitive events.

$$H1_{(A-B)} = \|W\|_{(A-B)}$$
$$H1 = \{H1_{(A-B)}\} \quad \forall (A-B) \in \ W$$

Where W is the set of all PEs appearing in the window from region SR A towards SR B.

$$H2_{(A-B)} = Avg((A-B).T) \quad \forall (A-B) \in W$$
$$H2 = \{H2_{(A-B)}\} \quad \forall (A-B) \ \in \ W$$

In the case that an Activity is repeated at different time in the video, we average the histograms extracted from each instance.

6.3. Activity Recognition

The activity recognition is performed in 2 steps. First, the PEs of DT_{Test} are calculated, the labels of the obtained PEs are changed to maintain similarity with DT_{Learn} activity models. The labels are changed, finding the alignment between DT_{Test} and DT_{Learn} topologies.

Second, similar histograms to the activity model are searched in the re-labeled PE sequence of DT_{Test}.

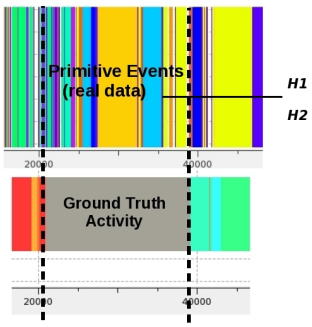


Figure 7. Activity model learning

6.4. Topology Alignment

To locate an Activity spatially, it is important to label the Slow Regions of DT_{Test} with the ones of the DT_{Learn}.
We use a relational graph matching algorithm [10], capable of computing pair matching between graphs of different number of nodes.
The method captures the local structural information (K-plet) by computing the K nearest neighbors for each SR, by inferring all possible configurations and by keeping the one maximizing the matching between the two graphs.
The matching between K-plets of DT_{Learn} and DT_{Test} SRs is performed using a *dynamic programming* based algorithm.
The consolidation of the local matches is done using *Coupled Breadth First Search (CBFS)* algorithm [10]. (CBFS) propagates the local (k-plets) matches simultaneously in both datasets.
We feed the algorithm with the SRs of DT_{Test} and DT_{Learn}. The matched SR pairs are used to re-label the DT_{Test} PEs sequence (i.e if the SR #4 of DT_{Learn} is similar to SR #1 of DT_{Test}, a primitive event $(1-4)$ of DT_{Test} is transformed to $(4-4)$).

6.5. Activity Search

We slide iteratively a temporal window W over the computed sequence of PEs of the testing video (DT_{Test}). Since the duration of the learnt activity is strict, we vary the size of W at each iteration. From the subsequence of PEs contained in W we extract two histograms $H1^*$ and $H2^*$ (in a similar manner than $H1$ and $H2$ are learnt for the activity modeling). We compute a similarity measurement ($dist_W$) between the model and the extracted histograms.

$$dist_W = dist_{H1*} + dist_{H2*} * k \tag{12}$$

$$dist_{H1*} = \sum \begin{cases} \|H1_{(A-B)} - H1^*_{(A-B)}\| \\ H1^*_{(C-D)} * t \\ H1_{(E-F)} * q \end{cases} \tag{13}$$

$$dist_{H2*} = \sum \begin{cases} \|H2_{(A-B)} - H2^*_{(A-B)}\| \\ H2^*_{(C-D)} * m \\ H2_{(E-F)} * s \end{cases} \tag{14}$$

$\forall (A-B) \in H1^* \cap H1$
$\forall (C-D) \in H1^* \wedge (C-D) \notin H1$
$\forall (E-F) \in H1 \wedge (E-F) \notin H1^*$
Where $H1$ and $H2$ stands for the activity model histograms, $H1^*$ and $H2^*$ are the histograms extracted from a time window W placed in the new dataset, $H1^*_{(A-B)}$ is the histogram value for the primitive event $(A-B)$, k is a utility factor (i.e. $k=0$ means that the duration of the PEs is not important to recognize an activity).
t, q, m, s are penalty factors (i.e. setting $s, q = 0$, implies that the PEs of the model that do not appear in the test histograms are not considered for the similarity measure, allowing outlier PEs in the activity recognition).
The topology alignment procedure allows to set different topology granularity for DT_{Test} and DT_{Learn}, thus we can set a higher granularity to DT_{Test} topology to filter out non shared logical regions.
The similarity measure used in the activity search algorithm could be extended to take into account the proximity of the recognized PEs in the global distribution.

7. Experiments

For experimentation we selected three of the shared activities between DT_{Test} and DT_{Learn}. The activities are: "Working in the kitchen","Working at the Table" and "Eating".

7.1. Learning Experiments

The parameters used are $H_{SLOW} = 120cm/s$, $Q_{STAY} = 4sec$. The number of clusters used for learning the DT_{Learn}'s scene topology is 8 (the resulting SRs -Slow Regions- are displayed in figure 10 (a)). We compute the PE sequence of DT_{Learn}, and we extract the activity models of the 3 selected activities. In *Table 1* the most relevant PEs of the "Working in the kitchen" activity histogram are displayed, showing high interaction between regions 1-3-5, corresponding to the kitchen area (figure 10).

3-1	1-3	3-3	1-5	1-1	5-1	5-5	3-5	...
7	6	6	6	4	4	2	2	...
149	153	165	177	624	219	122	158	...

Table 1. Working in the kitchen - first line: primitive Event type (i.e. transition between two SRs), second line: number of occurrences, third line: time spend for the PE

7.2. Activity Recognition Experiments

We compute the topology of DT_{Test} using 9 clusters (the number of clusters can be greater than the learnt

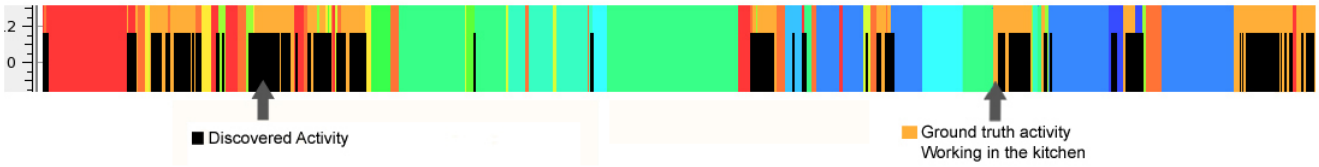


Figure 8. "Working in the kitchen" activity detection: the recognized activity intervals that best fit the model are colored in black. The "Working in the kitchen" ground truth activities are colored in orange.

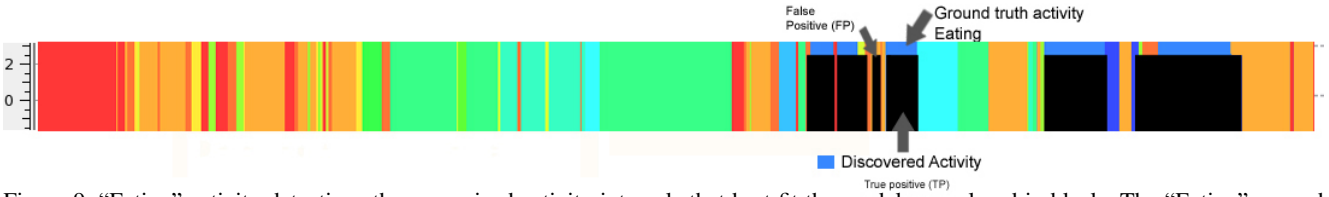


Figure 9. "Eating" activity detection: the recognized activity intervals that best fit the model are colored in black. The "Eating" ground truth activities are colored in orange.

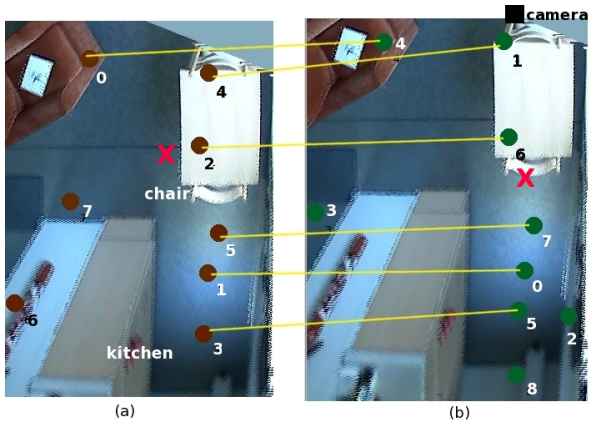


Figure 10. Topology correspondence (a) Learning dataset Topology (DT_{Learn}) (b) Testing Dataset Topology (DT_{Test}) - The regions where the persons in DT_{Test} and DT_{Learn} are eating are marked with "X"

clusters to allow some flexibility in the matching process), and we align DT_{Test} and DT_{Learn} topologies. The alignment find 6 SRs (figure 10 (b) yellow lines) shared between DT_{Test} and DT_{Learn} topologies. The shared SRs are relevant enough to recognize the selected activities.

We compute the PE sequence of DT_{Test}, and we search the DT_{Learn}'s modeled activities.

To validate the activity recognition, we use an activity ground truth (GT_{Test}). We search for the N histograms that better fit the model respectively for each activity in Table 2. The histograms represent the interval of time where the activity is recognized. We align the intervals with the activity ground truth (figure 8-9) and calculate the following performance measures:

TP = True Positives, number of detected activity intervals that overlap the ground truth activity.

TPT= True Positive Time, the time percentage of activity intervals that overlap the ground truth activity.

FDT = False Detection Time,the time percentage of activity intervals that do not overlap the ground truth activity for the whole video.

AT = Number of Ground Truth activity instances.

AD = Number of detected Ground truth activity instances.

The experimental results are displayed in *Table 2*. An example of recognized "Working in the kitchen" can be found in figure 8. All the activity instances are recognized, and most of the False Detection segments are of small size. An example of "Eating" is illustrated in figure 9. The time duration of the TPs is more accurate than the previous example. Is important to note that the activity is recognized even when the chair position is different between DT_{Learn} and DT_{Test} when the person is eating (see "X" in figure 10). This is because the selected clustering granularity merges all the slow points near the corner of the table in one SR, allowing the traslation of the chair.

Activity	TP	TPT	FDT	AT	AD
Working in the kitchen	26	74%	3%	16	16
Eating	5	100%	1%	5	5
Working at the Table	34	76%	2%	28	26

Table 2. Activity detection results.

The results show for the algorithm a good capability to recognize different types of activities (high/low PEs interaction and long/short time duration).

8. Conclusions

We propose a novel method to detect and recognize loosely constraint activities in unstructured scenes. We first propose a method to learn the scene logical regions (scene topology) in a unsupervised way.

We show that the learnt regions for two different individuals can be matched when the individuals are performing similar activities. We secondly propose a bridge (Primitive Event) that links automatically vision features (trajectories) and high level activities. We propose an activity search method capable of detecting long/short term activities based on Primitive Event histograms.
Future work include learning in an unsupervised way the model of the activities by taking into account a large training dataset containing a total of 14 monitored elderly people. Also we are working on improving the similarity measurement for the searching activity process by using a Bayesian characterization of the activities. Although that the tuning of the searching activity process parameter does not have a strong impact on the performance results we are still planning to optimize these parameters using Genetic Algorithm.

References

[1] S.Calderara, R. Cucchiara, and A. Prati. Detection of abnormal behaviors using a mixture of von mises distributions. In *IEEE AVSS 2007*.

[2] N. Anjum and A. Cavallaro. Multi-feature object trajectory clustering for video analysis. In *IEEE Transactions on Circuits for Video Technology*, pages 1555–1564, 2008.

[3] G. L. Foresti C. Piciarelli, C. Micheloni. Trajectory-based anomalous event detection. In *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1544–1554, 2008.

[4] R. Hamid, S. Maddi, A. Johnson, A. Bobick, I. Essa, and C. Isbell. A novel sequence representation for unsupervised analysis of human activities , artificial intelligence journal, accepted paper. 2008.

[5] Brendan T. Morris and Mohan M. Trivedi. Learning and classification of trajectories in dynamic scenes: A general framework for live video analysis. *Advanced Video and Signal Based Surveillance*, 2008.

[6] Weiming Hu, Xuejuan Xiao, Zhouyu Fu, and Dan Xie. A system for learning statistical motion patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(9):1450–1464, 2006. Fellow-Tan, Tieniu and Member-Maybank, Steve.

[7] Xi Li, Weiming Hu, and Wei Hu. A coarse-to-fine strategy for vehicle motion trajectory clustering. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 591–594, 2006.

[8] Adrian Hilton Thomas B., Moeslund and Volker Kruger. A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Image Underst.*, 104(2):90–126, 2006.

[9] D. Makris and T. Ellis. Learning semantic scene models from observing activity in visual surveillance. *SMC-B*, 35(3):397–408, June 2005.

[10] Sharat S. Chikkerur. Online fingerprint verification system - m.s. thesis - http://web.mit.edu/sharat/www/home.html. 2005.

[11] S. Khalid and A. Naftel. Classifying spatiotemporal object trajectories using unsupervised learning of basis function coefficients. In *VSSN '05: Proc. of Intl Workshop on Video Surveillance & Sensor Networks*, 2005.

[12] F. Porikli. Learning object trajectory patterns by spectral clustering. *Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on*, 2:1171–1174 Vol.2, June 2004.

[13] G. Antonini and J. Thiran. Trajectories clustering in ICA space: an application to automatic counting of pedestrians in video sequences. In *Advanced Concepts for Intelligent Vision Systems, ACIVS 2004, Brussels, Belgium*, Proc. Intl. Soc. Mag. Reson. Med. IEEE, 2004.

[14] S.G. Gong and T. Xiang. Recognition of group activities using dynamic probabilistic networks. In *ICCV03*, pages 742–749, 2003.

[15] Ivan Laptev and Tony Lindeberg. Space-time interest points. In *ICCV*, pages 432–439, 2003.

[16] N.M. Oliver, B. Rosario, and A.P. Pentland. A bayesian computer vision system for modeling human interactions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):831–843, Aug 2000.

[17] Ismail Haritaoglu, David Harwood, and Larry S. Davis. W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000.

[18] J. Owens and A. Hunter. Application of the self-organizing map to trajectory classification. In *VS '00: Proceedings of the Third IEEE International Workshop on Visual Surveillance (VS'2000)*, 2000.

[19] Aaron F. Bobick and Andrew D. Wilson. A state-based approach to the representation and recognition of gesture. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(12):1325–1337, 1997.