Trajectory Based Activity Discovery

Guido Pusiol Francois Bremond Monique Thonnat
Pulsar, Inria Sophia Antipolis, France
gtpusiol,fbremond,thonnat@sophia.inria.fr
http://www-sop.inria.fr/pulsar/

Abstract

This paper proposes a framework to discover activities in an unsupervised manner, and add semantics with minimal supervision. The framework uses 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 of Primitive Events. The proposed representation for primitive events aims at capturing small meaningful motions over the scene with the advantage of being learnt in an unsupervised manner. We propose the discovery of an activity using these Primitive Events as the main descriptors. The activity discovery is done using only real tracking data. Semantics are added to the discovered activities and the recognition of activities (e.g., "Cooking", "Eating") can be automatically done with new datasets. Finally we validate the descriptors by discovering and recognizing activities in a home care application dataset.

1. Introduction

The automatic discovery 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 [21]. 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 discovery and recognition of activities (e.g., "Cooking") by constructing an intermediate layer of Primitive Events in a completely unsupervised way.

The intermediate layer tries to capture the motion of the individual to perform basic tasks, using only minimal information (trajectories). Using visual information enables to reduce the complexity of systems that usually use numerous sensors to enrich the observation data [21].

To automatically model these primitive events first the scene topology is learnt in an unsupervised way, and meaningful transitions between topological regions are captured.

The composition of primitive events is very informative about the description of many activities. Thus we search for particular patterns within the primitive event layer to discover interesting activities. The patterns of primitive events are then used as generic activity models in order to recognize automatically the main activities for a new observed person.

These contributions are described in the third and eight sections. The process to build the scene topology is presented in the fourth section. The generation of primitive events and the discovery of activities are respectively described in the fifth, sixth and seventh sections. The paper concludes with validation experiments recognizing activities such as "Cooking" on home care monitoring application.

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, data-mining techniques can be applied on large video data. Recently particular attention has been focused 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.

Typical supervised methods such as [6, 4, 10] can build activity behavior models in a very accurate way. The problem is that they require big training datasets labeled manually.

The unsupervised methods generally include Neural Networks based approaches such as [8, 7, 15, 9]. They can represent complex nonlinear relations between trajectory features in a low-dimensional structure. These networks can be

trained sequentially and updated with new examples, but the complexity of the parametrization usually makes the networks grow and became useless after long periods of time.
Clustering approaches such as Hierarchical Methods [1, 11, 16] allow multi-resolution activity modeling by changing the number of clusters, but the clustering quality depends on the way to decide when a clusters should be merged or not. Adaptive Methods [17, 12], where the number of clusters adapts over time, makes possible on-line modeling without the constraint of maintaining a training dataset. In these methods it is difficult to initialize a new cluster preventing outlier inclusion. Other methods [19, 3] use dynamic programing based approaches to classify activity. These methods are quite effective when time ordering constraints hold. Hidden Markov Model based approaches such as [18, 20] capture spatio-temporal relations in trajectory paths, allowing high-level analysis of an activity, which is very suitable for detecting abnormalities. These methods are not very suitable for unstructured scenes and the adaptability in time is poor.
Several approaches on modeling scene context (topology) can be found in the literature [13], but just a few combine topology with activity interpretation. For instance, Morris and Trivedi [14] learn scene points of interest (POI) and model the activities between POIs with HMMs encoding trajectory points. This approach is suitable to detect abnormal activities and performs well when used in structured scenes (i.e., the usual trajectory paths are well defined, such as in a highway). The method requires activities to have time order constraints, and the topology is based on the trajectoy entry and exit zones. Hamid et al. [5] merge the scene topology and sensorial information, modeling sequences of events (n-grams) to discover and classify activities. The method requires manual specification of the scene. Most of the methods described above can be applied only in structured scenes (i.e., highway, traffic junction), and cannot really infer activity semantics. To solve these problems we propose a method capable of discovering loosely constraint activities in non structured scenes (i.e., an appartment), and we go up to semantic interpretation with minimal human knowledge.

3. Overview

We propose a method capable of understanding activities by combining small motion occurring in a scene. The method takes as input trajectories of different persons registered successively from the same camera viewpoint. Trajectories are obtained by a tracking algorithm [2] that builds a temporal graph of connected objects over time. We use calibrated cameras to calculate the moving objects 3D location on the ground plane. The moving objects are obtained by the classification (i.e., "person") of foreground image regions.
The proposed method is divided into two main parts: Activity Discovery and Activity Recognition. The Activity Discovery method takes trajectories (fig. 1 a) as input. These trajectories are processed using machine learning techniques to automatically discover and build activity models. The activity models can be manually labeled adding meaningful semantics (fig. 1 c). The outputs of the method are a topology of the scene (interesting regions shared by different individuals) and Activity Models (fig. 1 d).

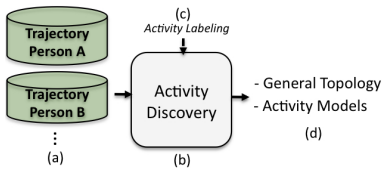

Figure 1. System Overview, (a) Input: Trajectory datasets from different persons. (b) Activity Discovery method (c) The user can add semantics to the learnt activities (d) Output: Learnt topology of interesting scene regions and learnt Activity Models.

A finer description of the Activity Discovery method can be found in fig. 2. The method starts by learning interesting scene regions for each individual. The set of interesting regions builds a scene topology (fig. 2 a) that aims at describing the spatial locations where an individual performs activities. The learnt topologies are fused modeling a scene General Topology (fig. 2 b), capable of representing stable interesting regions shared by most of the persons. On the other hand, the trajectories of each individual are cut into small meaningful segments -also known as tracklets- (fig. 2 c). The abstraction of trajectory segments using topology information builds Primitive Event descriptors. The sequence of Primitive Events (fig. 2 d) describes the motion of a person over the scene. From the Primitive Events sequence, particular subsequences (patterns) are extracted (fig. 2 e) to automatically discover interesting activities. The activities are modeled by performing statistics over the extracted patterns (fig. 2 f). The modeled activities are displayed to the users, who can label any activity they want (fig. 2 g). The Activity Recognition method is an extension of the Activity Discovery method and is explained in section 8.

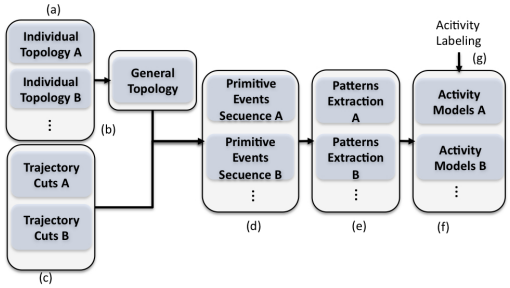

Figure 2. Stages of the Activity Discovery method

4. Scene Topology

Each individual when interacting with scene objects (e.g., "kitchen sink", "armchair") defines a set of regions characterizing a particular topology of the scene. The scene topology is a set of regions learnt through clustering of meaningful trajectory slow points.

4.1. Trajectory Slow Point

A trajectory slow point is a trajectory point that describe that the person is still. The speed of a trajectory point p_i is estimated by the object spatial distance walked within a fixed window of points, centered at p_i (fig. 3 a). Let T be a trajectory, where $T =< p_1, ..., p_n >$:

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

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

where H_{SLOW} is the minimum speed for a person to be considered still. To capture meaningful interaction between the person and the scene objects, the person has to remain still for a certain period of time. Thus we extract "segment of slow points" or $SSLP$ (fig. 3 b). Let S be a sequence of points $< p_{i-q}, ..., p_{i+q} >$, of a trajectory T, then:

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

where Q_{STAY} is a threshold that determines the least amount of time a person has to remain still. A $SSLP_i$ is represented spatially as a single point "scene slow point" (SSL_i) which is the average of the points contained in a $SSLP_i$. Also we define the length of a SSL_i related to a $SSLP_i$ to be the number of points in $SSLP_i$.

$$SSL_i = Avg\{p \,|p \in SSLP_i\} \quad (4)$$

$$SSL_i.length = |SSLP_i| \quad (5)$$

We calculate the set of SSLPs from the whole trajectory dataset of a person (fig. 4 a). From the set of $SSLPs$ we extract the $SSLs$ (fig. 4 b). We are interested into obtaining a single feature to describe spatially the several instances of time that a person stops at the same location. Thus we perform K-Means clustering over the set of SSLs (fig. 4 b). The number of clusters at this stage should be large enough to capture the scene logical regions that describe activity. Over-estimating the number of clusters is not a problem since the clusters are refined in a future step.

4.2. Individual Topology

An individual topology is defined as a set of Slow Regions (SR) for a single individual (fig. 4 c). Each SR is obtained by extracting information of a SSL Cluster (fig. 4 b).

First, the label and the spatial center of a $Cluster_{(i)}$ are given to the region SR_i:

$$SR_i = i \quad \text{(the label of } Cluster_{(i)})$$
$$SR_i.centroid = Cluster_{(i)}.centroid$$

Second, to describe temporal properties, two attributes are associated to SR_i: 1) a Gaussian function describes how often the region is visited. 2) another Gaussian function describes how much time a person stays still in the region.

$$SR_i.RevisitTime = (\mu_1, \sigma_1) \quad (6)$$

$$SR_i.UsageTime = (\mu_2, \sigma_2) \quad (7)$$

The SSLs are ordered by time of appearance ($timestamp_{start}$) in the dataset, then $\forall\ SSL \in Cluster_{(i)}$:

$$\mu_1 = \frac{\sum(SSL_q.timestamp_{end} - SSL_{q-1}.timestamp_{start})}{\#Cluster_{(i)}}$$

$$\mu_2 = (\sum SSL_q.length)/\#Cluster_{(i)}$$

where σ_1 and σ_2 stand for the standard deviation.

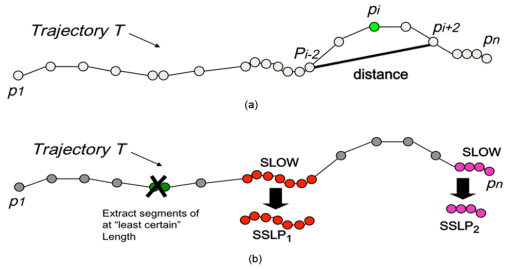

Figure 3. (a) Speed estimation of a trajectory point (b) Extraction of Strings of Slow Points (SSLP).

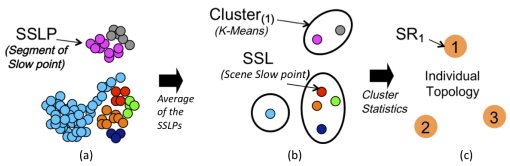

Figure 4. (a) Example of Segment of Slow Points (SSLPs) differentiated by color. (b) Example of Scene Slow Point (SSL) extracted from SSLPs maintaining its colors, and 3 clusters of the different SSLs. (c) Example of Slow Regions (SR) built from clusters of SSLs.

4.3. General Topology

To obtain a global understanding of the scene, the Individual Topology of different persons are merged in a single General Topology. The procedure is achieved by clustering (K-Means) over the SR centroids of all individual topologies. The number of clusters used for the topology merging corresponds to the abstraction level of activities. For example, for activity such as "Cooking" different clusters should describe the areas "sink" and "kitchen table".

An example of a General Topology can be found in fig. 5, where the Level is the number of clusters used.

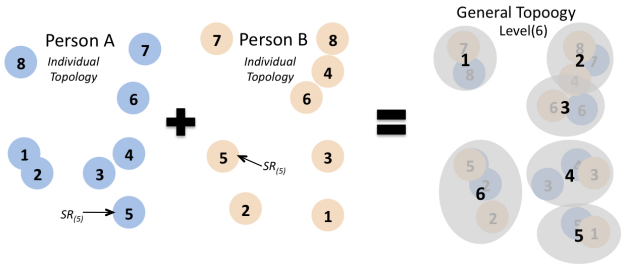

Figure 5. Example of general topology of Level (6). Obtained by clustering of the SRs of Person A and Person B.

5. Primitive Events

A Primitive Event (PE) is a descriptor that represents motion occurring between topology slow regions. First, meaningful motion is obtained by cutting a trajectory in segments (trajectory cut). Each segment aims at capture basic units of motion linked usually with a primitive activity (i.e., a person that is in movement stops to do something).
Second, each trajectory cut is spatially overlapped with a topology. From the overlapping, the information of the motion described by a trajectory cut over the topology is abstracted building a Primitive Event descriptor.

5.1. Cutting trajectories

A trajectory cut is a trajectory segment, that starts in the last point of a $SSLP_a$ and goes up to the last point of the next $SSLP_{a+1}$ where $SSLP_a$ and $SSLP_{a+1}$ are ordered by time of appearance in the trajectory. An example of trajectory cuts extraction is displayed in fig. 6. 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:

$$TC_i =< s_n, p_1, ..., p_m, q_1, ..., q_n > \quad (8)$$

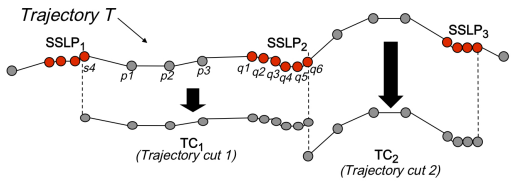

Figure 6. Example of two trajectory cuts (TC_1 and TC_2) extracted from a trajectory T.

5.2. Primitive Event Extraction

A PE is obtained by abstracting the information of the motion described by the trajectory cut over a topology. A primitive event is built by searching the nearest topology regions to the start/end points of the trajectory cut in the ground plane -3D- (see fig. 7 a). These regions are used to label a PE type. Also, the duration of a PE is extracted directly from the trajectory cut length. Then:

$$PE = (SR_{START} \rightarrow SR_{END})\ (Primitive\ Event\ type). \quad (9)$$

Where, given a trajectory cut $TC =< p_0, ..., p_n >$ and a topology $< SR_1, ..., SR_n >$:

$$SR_{START} = \arg\min_i(dist(p_0, SR_i)) \quad (10)$$

SR_{START} is the label of the nearest SR (Slow Region) of the scene topology to p_0.

$$SR_{END} = \arg\min_i(dist(p_n, SR_i)) \quad (11)$$

SR_{END} is the label of the nearest SR of the scene topology to p_n. Also,

$$(SR_{START} \rightarrow SR_{END}).duration = \frac{|TC|}{Frame\_Rate} \quad (12)$$

In figure 7 b, an example of a real primitive event extraction is displayed in 2D.

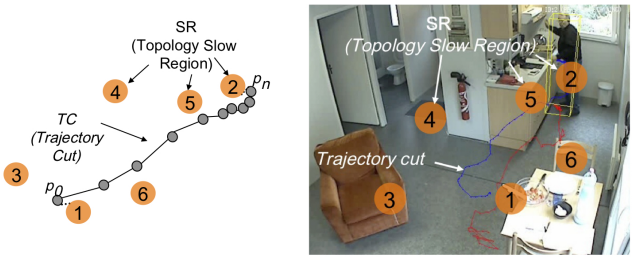

Figure 7. (a) Example of a Trajectory Cut (TC) over a Topology Slow Regions in the ground plane (b) Example of a primitive event $(1 \rightarrow 2)$ displayed in 2D, where $(1 \rightarrow 2).duration = 7sec.$ The blue and red lines are the tracked trajectories corresponding to the bottom and center of the person blob.

5.3. Primitive Events Sequence

For the whole trajectory dataset of a person we calculate the sequence of all primitive events (see fig. 9 a). An example of two PE sequences (Person A and Person D) is displayed in fig. 8. These sequences are obtained by merging the Individual Topology of both persons in a General Topology of Level(8). The PEs are ordered from left to right by their time of appearance in the video. Each color represents a PE type (same color means same PE type). The graph vertical axis displays the instance number of a PE, this is done to differentiate between subsequences of PE of the same type, e.g., several instances of a PE (fig. 8 a,b) compared with a long duration PE of a certain type (fig. 8 i). In fig. 8 can be noted that similar PE subsequences are describing similar activities between different persons, e.g., in (fig. 8 b,d,e) the activity "Eating", in (fig. 8 c,g) the activity "Cooking" and in (fig. 8 h,j) the activity "Sitting at the armchair".

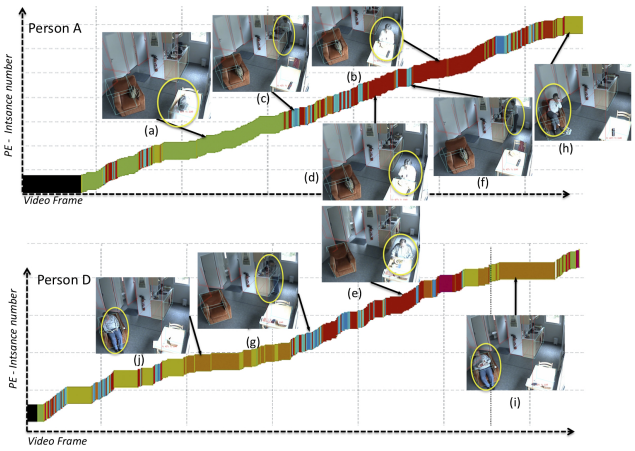

Figure 8. Primitive Event sequences of Person(A) and Person(D). Same color= same PE type.

6. Activity Discovery

The discovery of activities is performed by the extraction of particular subsequences from a sequence of primitive events (see fig. 9 a). We call these subsequences patterns. The modeling of an activity is obtained by performing statistics over an extracted pattern.

6.1. Pattern Extraction

First, we extract STATIC patterns, that describe activity occurring in only one topology region (e.g., "Sitting at the armchair"). These patterns are subsequences of PEs which all satisfy $SR_{START} = SR_{END}$.

Second, we extract MOVEMENT patterns. These can describe activity between two topology regions, e.g., "Cooking" usually involves PEs between different kitchen regions such as the "kitchen sink" and "kitchen table" regions.

Since the Primitive Events are ordered by time of appearance in a PE sequence, the MOVEMENT and STATIC patterns can be expressed with regular expresion notation. Let S to be a Primitive Events sequence, let A and B to be SRs then: A $MOVEMENT$ pattern is a maximal subsequence of S of the type:

$$\left((A \to A|B \to B)^*(B \to A|A \to B)^+(A \to A|B \to B)^*\right)^+ \tag{13}$$

A $STATIC$ pattern is a maximal subsequence of S of the type:

$$(A \to A)^+ \tag{14}$$

In (fig. 9 a), a Topology and an extracted sequence of Primitive Events is displayed. From the sequence, the MOVEMENT pattern extraction is displayed in (fig. 9 b), and the STATIC patterns extraction is displayed in (fig. 9 c).

6.2. Activity Models

We abstract the structure information of each extracted pattern by building two histograms. A first histogram (H1) fig. 9 e, captures the spatial properties of the pattern, counting the amount of instances of each PE type appearing in the pattern. A second histogram (H2) fig. 9 f, captures the amount of time used by the PE instances in the pattern.

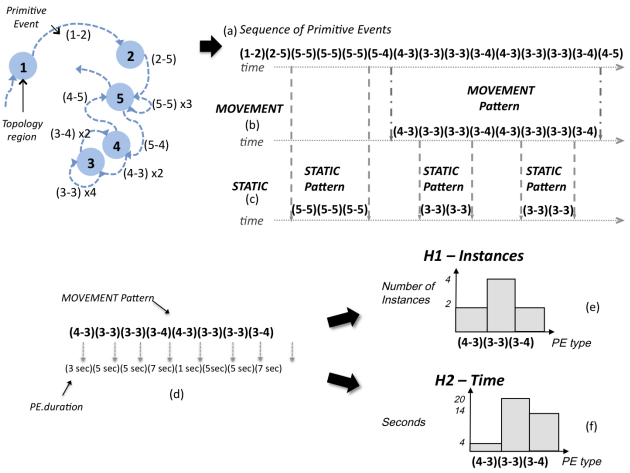

Figure 9. (a) Primitive Event sequence extraction, (b) MOVEMENT Pattern extraction, (c) STATIC pattern extraction, (e)(f) Histograms that model an activity.

7. Activity Similarity

The problem of finding similar activities automatically, is now reduced to calculating a distance ($dist$) between the histograms of two extracted patterns. We use a general distance measurement:

$$dist = dist_{instances} + dist_{time} * K \tag{15}$$

$$dist_{instances} = \sum \begin{cases} |H1_{(A\to B)} - H1'_{(A\to B)}| \\ H1'_{(C\to D)} * t \\ H1_{(E\to F)} * q \end{cases} \tag{16}$$

$$dist_{time} = \sum \begin{cases} |H2_{(A\to B)} - H2'_{(A\to B)}| \\ H2'_{(C\to D)} * m \\ H2_{(E\to F)} * s \end{cases} \tag{17}$$

$\forall (A \to B) \in H1' \cap H1$
$\forall (C \to D) \in H1' \wedge (C \to D) \notin H1$
$\forall (E \to F) \in H1 \wedge (E \to F) \notin H1'$

where $H1$ and $H2$ stand for the histograms of the first pattern, $H1'$ and $H2'$ are the histograms of the second patten, $H1'_{(A\to B)}$ is the histogram value for the primitive event $(A \to B)$, K is an utility factor. t, q, m, s are penalty factors. The distance parametrization is dependent on what the user is looking for. For example, setting $s, q = 0$, implies that the PEs of the first pattern that do not appear in the second pattern are not considered for the similarity measure, allowing similar activities to have outlier PEs. Also, if only the spatial motion is important, a $K = 0$ parameter should be configured. Parameters were chosen by hand, but we are planning to use non parametric distances (e.g., Earth Mover's Distance).

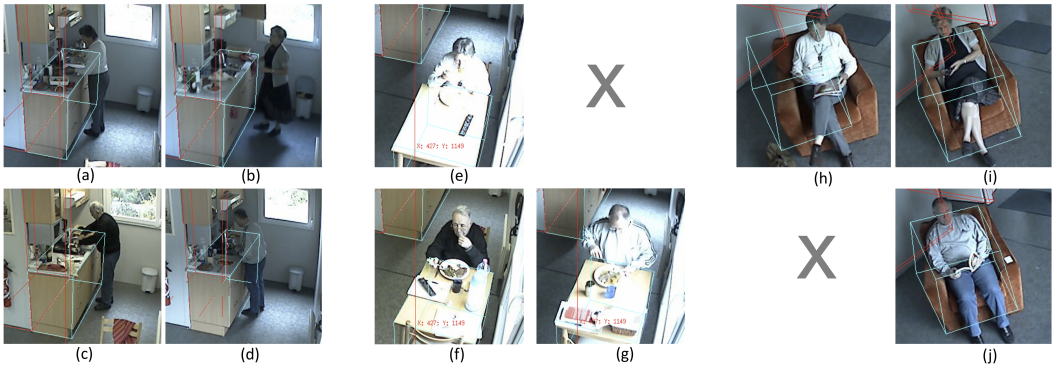

Figure 10. Shared activities by different persons, (a)(b)(c)(d): "Cooking", (e)(f)(g): "Eating", (h)(i)(j): "Sitting at the armchair". The "X" expresses that the activity is not present for the missing person.

8. Activity Recognition

The Activity Recognition Method is an extension of the Activity Discovery method, that uses learnt information to recognize activity in a new dataset. A description of the method stages is displayed in fig. 11.

The method takes as input a trajectory dataset of a new unseen person (fig. 11a), a learnt general topology (fig. 11 b) and learnt activity models (fig. 11 c). From the new dataset, the Trajectory Cuts and the Individual Topology (fig. 11 f,d) are calculated. A new General Topology (fig. 11 e) is built merging the learnt General Topology (fig. 11 b) and the Individual Topology (fig. 11 d) allowing to find a spatial correspondence between the learnt and the new Individual Topology. The Primitive Events sequence and the patterns (activity discovery) are extracted for the new person (fig. 11 g,h). Then, discovered activities are modeled (fig. 11 i) and the recognition is performed by measuring the similarity between the new models (fig. 11 i) and the learnt models (fig. 11 c). When a learnt activity model is labeled, the recognized activity takes the label as its semantic meaning.

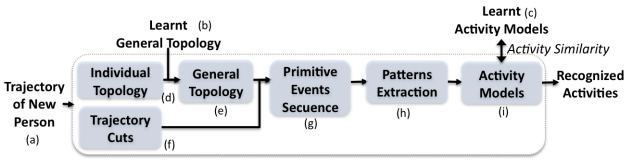

Figure 11. Stages of the Activity Recognition method

9. Experiments

For experimentation we use 4 datasets -Person A,B,C,D- (see fig. 10 a,b,c,d). Each dataset contains 4 hours of video, of elderly people living in an apartment. The videos are captured from 10 am to 2 pm.

The performance of the Activity Recognition method, directly depends in the Activity Discovery method. Thus we evaluate the Activity Recognition method to capture the performance of the Activity Discovery method.

We evaluate the Activity Recognition method using cross validation technique. The cross validation is performed using 3 dataset for training and discovering activities in the 4th dataset (total 4 experiments: one for each possible test person). The steps are the following: 1) Learn the General Topology of the 3 training datasets; 2) Discover activity in the training datasets; 3) Label some of the activities discovered; 4) Build a new General Topology between the learnt and the Individual Topology of the 4th person; 5) Discover activity of the 4th person; 6) Recognize the activities of the 4th person using the learnt ones; 7) Perform the evaluation of the recognized activities using an activity ground truth.

For recognition, at least one sample of the modeled activity (learnt) is required. Thus we limit ourselves to evaluate the recognition performance of activities that we know they are shared by at least two persons. We tagged 5 activities such as: "Cooking" (see fig. 10 a,b,c,d); "Sitting at the armchair" (see fig. 10 e,f,g); "Eating at position A" (see fig. 10 h,i,j); "Sitting at position B", "Going from the kitchen to the bathroom". For a scene description of the the location of position A-B, armchair, bathroom and kitchen, see fig. 12 a.

Aiming at a general approach, the parameters are set to be the same for all persons. We set the parameters using logical assumptions corresponding to the type of scene: $H_{SLOW} = 1meter/sec$; $Q_{STAY} = 3sec$, Individual topology of Level (12), General Topology of Level (10).

9.1. Performance Measures

For each dataset, a ground truth (GT) of activities is manually labeled (see fig. 13). The GT describes the instants where and activity begin and end. Each recognized activity is compared with the GT and the following measurements are extracted:

Let Act_i to be an activity learnt using Persons B,C,D and we recognize Act_i in Person A then:

GT$_A$: Amount of instances of Act_i appearing in the ground truth.

Detection$_A$: Amount recognized Act_i instances of the GT.

Missed$_A$: Amount of GT instances of Act_i not recognized.
Confusion$_A$: Amount of instances of Act_j recognized as Act_i $(j \neq i)$.
GTDuration$_A$: Time duration of all instances of Act_i in the GT.
DetectionDuration$_A$: Time duration of the recognized instances of Act_i corresponding to Act_i in the GT.
ConfusionDuration$_A$: Time duration of instances Act_j recognized as Act_i, where $j \neq i$.

9.2. Results

Since cross validation technique is used, the results of Tables 1 and 2 correspond to the accumulation of the measurements described above for the 4 experiments. The extracted metrics show a good performance of the algorithm to recognize most of typical home care application activities. It is interesting to see the recognition results in fig 13, where for "Cooking" the recognized segments are colored by similarity with activity models: this gives coarse idea of how an activity is performed. Also, note that for "Sitting at the armchair" a single learnt activity model is enough for the recognition.

Activity	GT	Detection	Missed	Confusion
Sitting at the armchair	12	11	1	4
Cooking	70	47	6	8
Eating at position A	16	13	3	4
Sitting at position B	15	13	2	6
Going from the kitchen to the bathroom	4	3	1	1

Table 1. Results of instances recognition of activities.

Activity	GT Duration	Detection Duration	Confusion Duration
Sitting at the armchair	121 min	117 min	4 min
Cooking	115 min	93 min	7 min
Eating at position A	65 min	54 min	8 min
Sitting at position B	143 min	131 min	13 min
Going from the kitchen to the bathroom	3 min	2 min	less than1min

Table 2. Results of temporal metrics of recognized activities.

We also would like to show the reason for the algorithm errors displayed in the Tables 1 and 2. Most of the errors are produced by illumination changes (fig. 12 b), background integration and dynamic objects changing location. These problems should be addressed in a lower vision level. Other errors are confusions produced by indistinguishable motion (fig. 12 d) between activities at the same location, and camera perspective issues (fig. 12 c). Solving these problems is part of our future work.

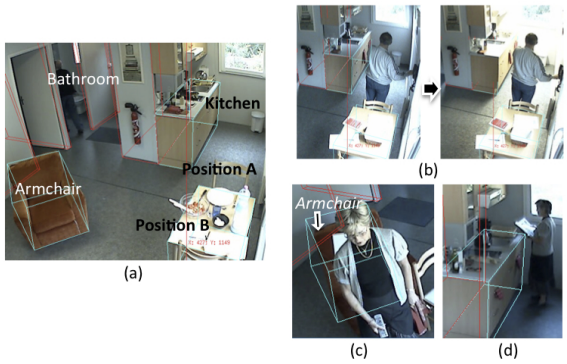

Figure 12. (a) Example of scene logical regions. (b) The person turns on the kitchen light producing a change of the illumination that leads to a tracker mistake. (c) The camera perspective confuses the person "Standing" as if it is "Sitting at the armchair", (d) The motion of the person while it is "reading a magazine" at the kitchen is confused with the activity "Cooking".

9.3. Conclusions

We proposed a method to discover and recognize activities loosely constrained, in unstructured scenes. The contributions are summarized as: An algorithm to learn scene interesting regions (Topology); a data structure that is able to capture small spatial translations over the scene (Primitive Event); a method to combine the small tasks to discover activities without human intervention (Patterns); a method to recognize activities in an unseen dataset. These activities can be linked either to learnt activities in an unsupervised way or to manually annotated labels. We evaluated the performance of the method, quantifying the potential of Primitive Events to discover and recognize activity. The proposed method can detect most of the interesting activities in a home-care application. Our future work is focused into learn and recognize finer activities adding vision information to the system (e.g., action and shape descriptors) to solve situations as the described in fig. 12 c,d.

References

[1] 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. 2

[2] A. Avanzi, F. Brémond, C. Tornieri, and M. Thonnat. Design and assessment of an intelligent activity monitoring plat-

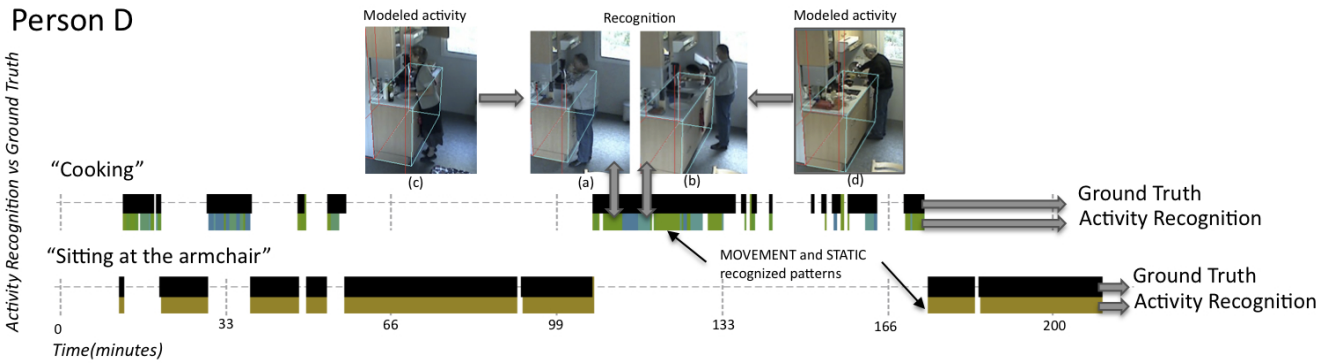

Figure 13. Examples of Ground Truth (Black lines) vs Activity Recognition (Colored lines) of Person D in the video timeline. Bottom: "Sitting at the armchair" activity. Top: "Cooking" activity. The different colors in the recognition timeline corresponds to the different patterns that are modeled as part of the activity. For instance, (a) is recognized thanks to (c) taken from Person B. Similar it is between (b) and (d) taken from Person C. Note that within an activity it can be distinguished the "way" an activity is performed

form. *EURASIP J. Appl. Signal Process.*, 2005(1):2359–2374, 2005. 2

[3] A. F. Bobick and A. D. Wilson. A state-based approach to the representation and recognition of gesture. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(12):1325–1337, 1997. 2

[4] J. W. Davis and A. F. Bobick. The representation and recognition of human movement using temporal templates. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference*, 0:928, 1997. 1

[5] 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. 2

[6] I. Haritaoglu, D. Harwood, and L. S. Davis. W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000. 1

[7] W. Hu, X. Xiao, Z. Fu, and D. 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. 1

[8] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. In *BMVC '95: Proceedings of the 6th British conference on Machine vision (Vol. 2)*, pages 583–592, Surrey, UK, UK, 1995. BMVA Press. 1

[9] 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. 1

[10] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, pages 432–439, 2003. 1

[11] X. Li, W. Hu, and W. 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. 2

[12] D. Makris and T. Ellis. Learning semantic scene models from observing activity in visual surveillance. *SMC-B*, 35(3):397–408, June 2005. 2

[13] S. Mckenna, , S. J. Mckenna, and H. Nait-charif. Learning spatial context from tracking using penalised likelihoods. In *In International Conference on Pattern Recognition*, pages 138–141, 2004. 2

[14] B. T. Morris and M. M. Trivedi. Learning and classification of trajectories in dynamic scenes: A general framework for live video analysis. *Advanced Video and Signal Based Surveillance*, 2008. 2

[15] 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. 1

[16] L. Patino, H. Benhadda, E. Corvee, F. Bremond, and M. Thonnat. Extraction of activity patterns on large video recordings. *Computer Vision, IET*, 2(2):108–128, June 2008. 2

[17] C. Piciarelli and G. L. Foresti. On-line trajectory clustering for anomalous events detection. *Pattern Recogn. Lett.*, 27(15):1835–1842, 2006. 2

[18] 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. 2

[19] S.Calderara, R. Cucchiara, and A. Prati. Detection of abnormal behaviors using a mixture of von mises distributions. In *IEEE AVSS 2007*. 2

[20] T. Xiang and S. Gong. Video behavior profiling for anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:893–908, 2008. 2

[21] N. Zouba, F. Bremond, and M. Thonnat. Multisensor fusion for monitoring elderly activities at home. In *IEEE International Conference on Advanced Video and Signal based Surveillance.*, Genoa, Italy., September 2009. 1