Unsupervised Discovery, Modeling and Analysis of long term Activities

Guido Pusiol, Francois Bremond, and Monique Thonnat

Pulsar, Inria - Sophia Antipolis, France

Abstract. This work proposes a complete framework for human activity discovery, modeling, and recognition using videos. The framework uses trajectory information as input and goes up to video interpretation. The work reduces the gap between low-level vision information and semantic interpretation, by building an intermediate layer composed of Primitive Events. The proposed representation for primitive events aims at capturing meaningful motions (actions) over the scene with the advantage of being learned in an unsupervised manner. We propose the use of Primitive Events as descriptors to discover, model, and recognize activities automatically. The activity discovery is performed using only real tracking data. Semantics are added to the discovered activities (e.g., “Preparing Meal”, “Eating”) and the recognition of activities is performed with new datasets.

1 Introduction

More than 2 billion people will turn over 65 year old by the year 2050. It is of crucial importance for the research community to help aging adults live independently for longer periods of time. The transition from their homes to new and unknown environments (i.e. an assisted living facility) add stressors that deteriorate theirs mind, memory and body. If we can keep the elders in their own homes over longer periods of time, they are in an environment that they know and trust so they can have a greater confidence leading to better quality of live.

The understanding of daily activities is the key to help solve the problem and is a topic that remains open. In the literature the computational approaches assume usually prior knowledge of the activities and the environment. This knowledge is used explicitly to model the activities in a supervised manner. In video surveillance the systems produce large quantities of data and it becomes almost impossible to continually monitor these data sources manually. It is of crucial importance to build computer systems capable of analyzing human behavior with 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 scene. Nowadays 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 to the discovery and recognition of long term activities (e.g. “Eating”) by constructing an intermediate layer of Primitive Events in a completely unsupervised way.

The intermediate layer aims at capturing the motion of the individual to perform basic tasks, using only minimal information (person position and dynamics). The use of small amounts of information allows the fast analysis of large amount of data. The advantage of using visual information is that it is captured using non-invasive sensors and enables to reduce the complexity of systems that use numerous sensors to enrich the observation data [19].

To automatically model the Primitive Events: First, the human actions are learned in an unsupervised way. Second, scene contextual information is learned capturing meaningful scene regions. Third, the primitive events are built by merging the actions and the scene information.

The composition of primitive events is very informative about the description of many activities. Thus, we search for particular sequences within the primitive event layer to discover interesting activities. The discovered activities are used to build generic activity models and the modeled activities are recognized in new unseen video datasets.

This paper is divided as follows: In the third section we explain how actions are learned, in the fourth section how the scene contextual information is obtained, in the fifth section how actions are abstracted to primitive events and how to combine the primitive events to discover and model activities, in the sixth section the activity recognition procedure is explained and in the seventh section we evaluate the approach in home-care applications.

2 Related Work

The advances made in the field of object tracking allow data-mining techniques to be applied to large video data. Recently particular attention has been focused on the object trajectory information over time to understand long term activities. Trajectory-based methods to analyze activity can be divided in two groups, supervised and unsupervised.

Typical supervised methods such as [7, 11, 5] can build activity models in a very accurate way. The problem is that they require big training datasets labeled manually.

The unsupervised methods include Neural Networks based approaches such as [9, 8, 13, 10]. 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 become useless after long periods of time.

Clustering approaches such as Hierarchical Methods [1] allow multi-resolution activity modeling by changing the number of clusters, but the clustering quality depends on the way to decide when clusters should be merged or not. Adaptive methods [14], where the number of clusters adapts over time, make on-line modeling possible without the constraint of maintaining a training dataset. In these methods it is difficult to initialize a new cluster preventing outlier inclusion. Other methods [17, 2] use dynamic programing based approaches to classify activitivities. These methods are effective when time ordering constraints hold.

Hidden Markov Model (HMM) based approaches such as [15] capture spatio-temporal relations in trajectory paths, allowing high-level analysis of an activity, which is suitable for detecting abnormalities. These methods require prior domain knowledge and their adaptability in time is poor.

Morris and Trivedi [12] 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. if the usual trajectory paths are well defined, such as on a highway). But the method requires activities to have time order constraints. Also [6] merges the scene POIs and sensorial information. But the method requires a 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 an approach that is suitable to unstructured scenes and which is the first to combine local and global descriptors to recognize long term activities.

3 Actions

To understand activities, we propose first to learn the actions that compose them by cutting a video into meaningful action segments. Each segment aims at capturing a person's action such as "standing up". We mark the beginning and ending of a segment by detecting the person's change of state (motion/static). From a video datafile we obtain a sequence of action segments. At each segment we compute the person's main dynamics by clustering meaningful trajectories. Finally, we build $Action$ descriptors that capture the global and local motion of a person in an action segment.

3.1 Global Position and Speed

We compute the person position at each frame by using a person tracker. The position is given to a linear Kalman-filter $(K1)$. At each new frame the prediction of $K1$ is averaged with the new position observation (obs) obtaining a smoothed trajectory (pos):

$$pos_{frame_i} = Avg(obs_{frame_i}, K1(obs_{frame_{i-1}}))$$

The $speed$ of a person in a new frame, is computed by averaging the prediction of another Kalman-filter $(K2)$ and the real speed observation $(sobs)$ in the new frame:

$$speed_{frame_i} = Avg(sobs_{frame_i}, K2(speed_{frame_{i-1}})$$

3.2 Action Segments and Local Dynamics

An **Action Segment** starts with a person's change of state and ends with the next change of state (motion/static). The changes of state are computed sequentially by thresholding the person's $speed$ at each frame.

Local Dynamics are a set of short trajectories describing the motions in an action segment. To compute these trajectories, the algorithm starts by placing 500 KLT points [18] at the first frame of the action segment and tracks them [3] until the last frame. The resulting set of KLT trajectories is numerous and in long action segments noisy trajectories could appear. To filter the noise out we extract KLT trajectories where their start/end points are not far from the global position trajectory start/end points.

Several KLT trajectories could be describing the same motion; we cluster the KLT trajectories using Mean-Shift algorithm [4] to obtain the main Local Dynamic trajectories. Mean-Shift is performed using the entry/exit points of the KLT trajectories to avoid the problem of clustering different trajectory lengths. The advantage of Mean-Shift is that it detects the number of clusters automatically, and filters out small clusters.

In Figure 1 displays a sequence of action segments with the computed Local Dynamics, it can be noticed how the small movements of the person are captured and that the resulting number of Local Dynamics is compact and descriptive.

Other descriptors have been tried (SIFT, SURF), they perform similarly to KLT but with much slower computational speed, while with KLT we process in real time.

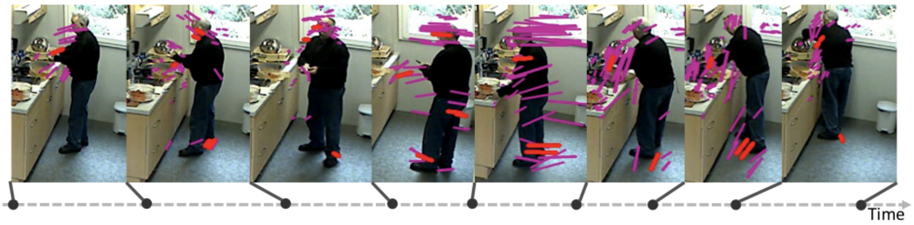


Fig. 1. From left to right a sequence of action segments with the computed KLT trajectories (pink) and Local Dynamics (red) after Mean-Shift clustering.

3.3 Action Descriptors

An $Action$ is a descriptor that captures global and local information of the trajectories in an action segment:

Globally: $Action_{posStart}$ and $Action_{posEnd}$ are the global person's position at the start/end frames of an action segment.

Locally: $Action_{Length}$ is the average length of the Local Dynamic trajectories and $Action_{Angles}$ is an histogram of the directions of the Local Dynamic trajectories normalized to $\{0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°\}$.

4 Scene Context

In this approach, no information about the scene is known. We learn a scene model composed by scene regions in order to locate actions spatially. The type of regions we are interested in are those where the individual interacts with the scene objects (i.e.,"armchair"). The set of learned regions is called a topology and it is learned by clustering trajectory points.

4.1 Learning a Topology

To build a topology we use the $Action_{posStart}$ and $Action_{posEnd}$ spatial points from a sequence of actions. These points are features describing the locations where the changes of state occur and describe the locations of interaction with the scene. Let $\langle Action_i \rangle$ be a sequence of actions. The set of $InterestPoints$ used is:

$$InterestPoints = \{Action_{i.posStart}\} \cup \{Action_{i.posEnd}\}$$

We perform K-Means clustering over $InterestPoints$. The number of clusters selected represents the level of abstraction of the topology, where lower numbers imply wider regions. Each cluster defines a Scene Region (SR). Finally, we denote $Topology_{levelN} = \{SR_0 ... SR_N\}$, where each SR_i is labeled with a number for later use.

4.2 Scene Model

A scene model is composed by 3 topologies. They aim at describing coarse, intermediate and specific scene regions. Figure 2 displays 3 topologies composing the model of a scene that we use for experimentation.

5 Activities

In this section we explain how to combine actions and scene contextual information to discover and model activities. First, we build activity descriptors named Primitive

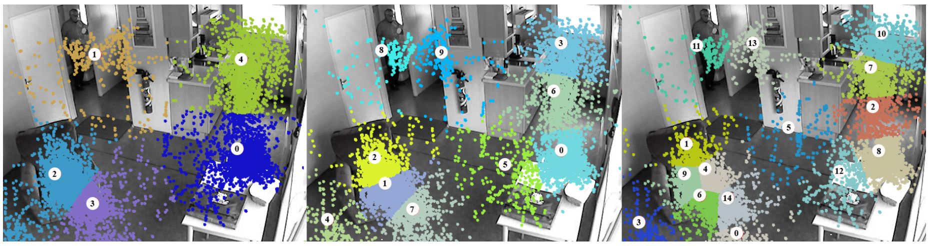


Fig. 2. Computed scene model corresponding to HOME-CARE dataset. From left to right the topologies of level 5, 10 and 15 are displayed. The labeled white dot represent the Scene Region center and the surrounding points the cluster members.

Events that capture an $Action$'s information over the scene. Second, we compute Primitive Event sequences of different levels of abstraction. Third, we combine the sequences to discover activities. And fourth, a discovered activity is modeled to be used by an activity recognition procedure.

5.1 Primitive Events

A Primitive Event (PE) is a descriptor that normalizes the global information of an $Action$ using a scene model. Suppose an $Action$ and a $Topology$ then the PE resulting from $Action$ is defined by its type as:

$$PE = (START \rightarrow END) \ \ (PE \, type).$$

where $START$ and END is the label of the nearest SR (Scene Region) of $Topology$ to $Action_{posStart/posEnd}$ respectively.

$$START = \arg\min_{i}(dist(Action_{posStart}, SR_i))$$

The $Action$ local descriptors are copied to the PE for later use.

$$(START \rightarrow END)_{Angles/Length} = Action_{Angles/Length}$$

5.2 Primitive Events Sequence

From a sequence of $Actions$, three Primitive Event sequences are computed. One for each $Topology_{Level}$ of a scene model. The motivation of having 3 levels of abstraction of PEs is that with the same set of descriptors, activities of different semantical abstraction levels can be discovered (e.g. “in the kitchen” and “at the kitchen sink”).

5.3 Activity Discovery

Independently for each PE sequence described in the previous section, we extract particular subsequences that describe activity. We are interested in two types of subsequences, denoted SPOTTED and DISPLACED.

SPOTTED describes activity occurring within a single topology region (e.g. “Reading in the Armchair”). These are composed by PEs of the same type.

DISPLACED describes activity occurring between two topology regions (e.g. “from Bathroom to Table”). These are composed by a single PE.

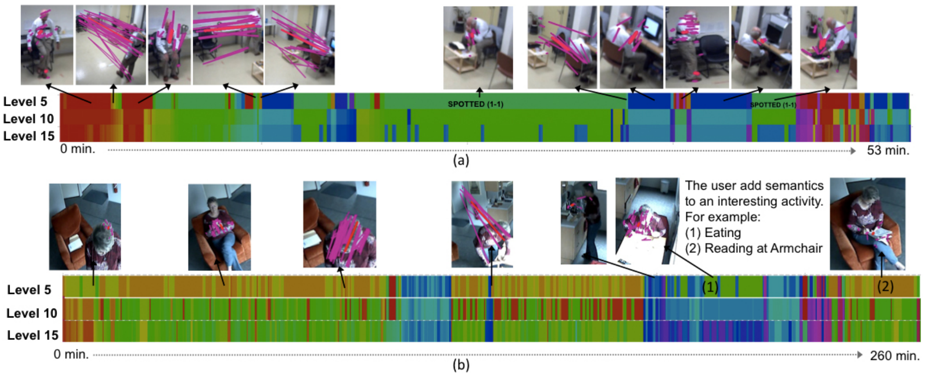
Using regular expressions, a $SPOTTED_{A-A}$ is a maximal subsequence of the PEs sequence of the type:

$$(A \rightarrow A)^{+} \qquad (1)$$

A $DISPLACED_{A-B}$ is a single PE of the type:

$$(A \to B), A \neq B \tag{2}$$

The discovered SPOTTED and DISPLACED subsequences are presented to the user as displayed in Fig. 3. The user labels the subsequence that represents an interesting activity at any of the 3 abstraction levels. Adding a label to a subsequence SPOTTED or DISPLACED defines an ACTIVITY SPACE that contains the Primitive Events used to model the activity. An example of how an ACTIVITY SPACE is built is displayed in Fig. 4, where we use the 3 topologies displayed in Fig. 2 to represent a configuration of PE sequences. The example shows how the SPOTTED and DISPLACED subsequences are computed and examples of the ACTIVITY SPACEs defined by labeling as "Preparing Meal" $SPOTTED_{4-4}$ and as "In kitchen table" $SPOTTED_{6-6}$.



The user add semantics to an interesting activity. For example:
(1) Eating
(2) Reading at Armchair

Fig. 3. Activity Discovery of 2 datasets: HOSPITAL (a) and HOME-CARE (b). The scene model used for (b) is displayed in Fig. 2. The colored segments correspond to DISPLACED and SPOTTED subsequences, where the same color is the same subsequence type. For example, SPOTTED(1-1) labeled at the abstraction level 5 (a) corresponds to activity of the person in the chair region. The displayed images are representative actions of the discovered activities.

5.4 Activity Model

An $Activity$ is modeled by 3 histograms (H_5, H_{10}, H_{15}) and a variable $Activity_{Length}$. Where H_l captures the information of the PEs sequence of $Level_l$ contained in an ACTIVITY SPACE. H_l is an histogram of 2 dimensions. The first coordinate $(globalfeature)$ is the type of a Primitive Event $(S \to E)$. The second coordinate $(localfeature)$ is an angle value θ. The count is the accumulation of θ of the primitive events of type $(S \to E)$ appearing in the PEs sequence of $Level_l$ of the ACTIVITY SPACE.

$$H_l(S \to E, \theta) = \sum (S \to E)_i.Angles(\theta) \tag{3}$$

The $Activity_{Length}$ is the average length $(S \to E)_{Length}$ of the Primitive Events appearing in the ACTIVITY SPACE.

6 Activity Recognition

For a new unseen video dataset, we aim at recognizing modeled activities in an unsupervised way. Suppose we have an $Activity$ as well as the learned scene model used

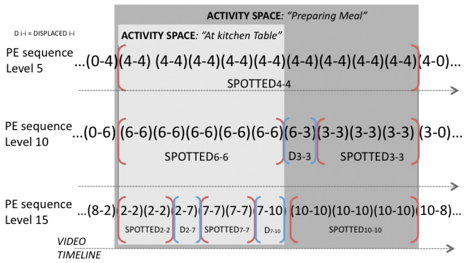


Fig. 4. Example of Activity Discovery sequences. Each layer represents a PE sequence at a level of abstraction. The brackets show the computation of SPOTTED and DISPLACE subsequences, and the ACTIVITY SPACEs are defined by labeling a SPOTTED or DISPLACE subsequence.

for modeling $Activity$. We are interested in finding a set of candidate activities that are similar to the modeled one. We explain the steps we use to find candidate activities in a new video:

First, the sequence of actions is computed as described in Section 3.

Second, the Primitive Event sequences are computed, as described in Section 5.2. The difference is that this time we do not compute a new scene model, instead we use the learned $scene\_model$. This way, the **PEs of the new video match spatially (PE type) with the PEs used for learning** $Activity$.

Third, the activity discovery process is performed as described in Section 5.3. From the computed set of SPOTTED and DISPLACED subsequences, those that match the subsequence used for labeling $Activity$ are selected.

For example, in Figure 2 we label $SPOTTED_{4-4}$ to model "Preparing Meal". For the new video, all $SPOTTED_{4-4}$ appearing at $Level_5$ are selected.

Fourth, the algorithm computes an ACTIVITY SPACE for each SPOTTED or DISPLACED selected in the previous step. From each ACTIVITY SPACE a candidate $Activity'$ is modeled as described in Section 5.4.

Fifth, because of the previous steps, a modeled $Activity$ and a candidate $Activity'$ have a global spatial correspondence. But this does not ensures that both activities are the same (i.e. two different activities may take place at the same spatial location). To measure the similarity we compute $score_{Length}$ and $score_{Histogram}$ and we compare the values to thresholds $T1$ and $T2$. To obtain a binary recognition, an $Activity'$ is the same as $Activity$ if the following statement is $true$:

$$score_{Length} < T1 \wedge score_{Histogram} < T2$$

Activity Similarity: We propose a distance that measures the similarity of all activity descriptors (local and global) by computing 2 scores between the model $Activity$ and the candidate $Activity'$.

The $score_{Length}$ measures the similarity length of the local dynamics:

$$score_{Length} = abs(Activity_{Length} - Activity'_{Length})$$

The $score_{Histogram}$ measures the similarity of the spatial position and local dynamic angles. This score is computed at the different levels of abstraction (capturing the sub-activities similarity) by comparing the 3 histograms of $Activity$ (H_5, H_{10}, H_{15}) with the 3 histograms of $Activity'$. We experimented different similarity measures for multidimensional histograms and finally adopted Earth Movers Distance (EMD):

$$score_{Histogram} = \sum EMD(H_i, H_i')$$

Thresholds: The recognition thresholds $T1$ and $T2$ are learned using the information of the modeled activities. Let $Model_1$... $Model_i$ be of the same activity, we calculate the mean $score_{Length}$ and $score_{Histogram}$ of all combinations as well as their standard deviation σ_1 and σ_2 . Then $T1$ and $T2$ are defined as:

$$T1 = Average(score_{Length}) + 2 * \sigma_1$$

$$T2 = Average(score_{Histogram}) + 3 * \sigma_2$$

7 Experiments

For experimentation we use videos of 2 different scenes: HOME-CARE and HOSPITAL datasets. Each video contains a single person and are recorded using a monocular video camera (640 x 480 pixels of resolution). HOME-CARE contains 7 elderly people performing non-guided activities in an apartment (in total 24 hours of video). HOSPITAL contains 4 videos of patients performing guided and non guided activities in a hospital room (3 hours of video). The last dataset is currently being used to study Alzheimer's disease symptoms and the protocol of the guided activities is described by Romdhane et al. [16].

From the discovered activities (i.e. Fig. 3) we label activities shared by most persons. They are selected using DISPLACED and SPOTTED subsequences, where the last ones are the most challenging because of possible activity confusions. For example, "Balance" and "Up/Down" are exercises for measuring the person's stability, both take place same location. The set of labeled activities is displayed in Tables 1, 2.

7.1 Evaluation

The Activity Recognition method depends on the Activity Discovery method, therefore the evaluation of the first one reflects the quality of the discovery procedure.

We evaluate the activity recognition method using cross validation technique. The evaluation is performed recognizing activities in a test video by learning the scene and activity models from the remaining videos. For example, in HOME-CARE, to recognize activities of person G, we compute the scene and activity models using the videos of persons A,B,C,D,E,F. In total 6 experiments are performed (one for each test video).

Performance measurements: For each dataset an activity ground truth (GT) is manually labeled. The GT describes the intervals of time when an activity begins and ends. The Activity Recognition method returns the intervals of time where an activity is recognized. Each recognized activity instance is compared with the GT and the following measurements are extracted:

True Positive (TP): Number of activity instances correctly recognized.

False Positive (FP): Number of recognized instances not appearing in the GT.

False Negative (FN): Number of instances appearing in the GT not recognized..

Recognition Time (RT): Percentage of time the activity is recognized, over the GT duration of the activity.

False Recognition Time (FT): Percentage of time the activity is recognized while it is not occurring in the GT, over the time the activity is recognized.

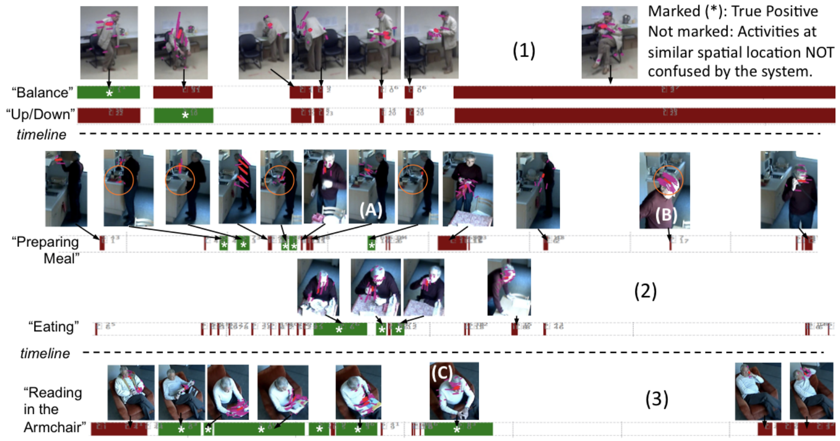

Marked (*): True Positive
Not marked: Activities at similar spatial location NOT confused by the system.
(1)
"Balance"
"Up/Down"
timeline
"Preparing Meal"
(A)
(B)
(2)
"Eating"
timeline
"Reading in the Armchair"
(C)
(3)

Fig. 5. Marked with (*) are the recognized segments (TP) of the activities: (1) "Balance" and "Up/Down"; (2) "Preparing Meal" and "Eating"; (3) "Reading at the Armchair". The activities are aligned in time. Not marked segments are other -different- activities occurring at the same spatial location not matching with the model. At the top, images representing characteristic actions of the activities. (A) is a False Negative due to lack of motion; (B) is an example of how local motion occurs at the "Preparing Meal" location, but there is no global position matching; (C) is a False positive due to similar motion and global position with the activity model.

Results: Table 1 and Table 2 display the recognition results. In both datasets the method has a very good performance. The FP occurs when the motion of the person while doing different activities is similar and the FN because of the lack of motion. The FT occurs because a person stop an activity without changing of place (i.e. at the end of Eating stays still for a while). To illustrate the complexity of the recognized activities we display some results graphically in Fig. 5.

Activity	TP	FP	FN	RT	FT
Balance	3	0	0	100%	1%
Up/Down	3	0	0	100%	4%
Reading at the table	10	1	1	95%	3%
Preparing Coffe	7	1	0	88%	5%
At the Computer	6	1	0	91%	4%
Excercice 1	3	0	0	99%	2%
Excercice 2	3	0	0	99%	1%

Table 1. Recognition results of the selected activities for HOSPITAL dataset.

Activity	TP	FP	FN	RT	FT
Eating	31	1	0	97%	7%
Reading in the Armchair	24	4	0	92%	11%
Preparing Meal	52	6	3	83%	6%
Standing at Armchair	11	2	0	95%	5%
Sitting at Eating place	8	0	1	99%	2%
Inside the bathroom	14	2	0	82%	7%
Armchair to Table	32	4	0	96%	1%
Armchair to Kitchen	15	1	0	98%	3%

Table 2. Recognition results of the selected activities for HOME-CARE dataset.

8 Conclusions

We propose a method to discover and recognize long term activities loosely constrained, in unstructured scenes. The insight of this paper is that it is the first time a complete framework links from the pixel level to complex semantics ("Eating"), using global and local features. Other approaches use either local or global features and the type of activities recognized can be considered as actions (sitting down in a chair).

The contributions are summarized as: An algorithm to learn a scene context (Activity Model); a data structure that combines global and local descriptors (Primitive Events); a method to combine small tasks to discover activities automatically; a method to recognize activities in new datasets. The evaluation results show that it can be used to study activities in home care applications and to perform fast and reliable statistics that can help doctors to diagnose diseases such as Alzheimer. Our future work is going to be the the extension of the approach to perform on-line activity recognition.

References

1. Antonini, G., Thiran, J.: Trajectories clustering in ICA space: an application to automatic counting of pedestrians in video sequences. In: ACIVS 2004. Proc. Intl. Soc. Mag. Reson. Med, IEEE (2004)
2. Bobick, A.F., Wilson, A.D.: A state-based approach to the representation and recognition of gesture. IEEE Trans. Pattern Anal. Mach. Intell. 19(12), 1325–1337 (1997)
3. Bouguet, J.Y.: Pyramidal implementation of the lucas kanade feature tracker description of the algorithm (2000)
4. Georgescu, B., Shimshoni, I., Meer, P.: Mean shift based clustering in high dimensions: A texture classification example. 9th ICCV pp. 456–463 (2003)
5. Gong, S., Xiang, T.: Recognition of group activities using dynamic probabilistic networks. In: ICCV03. pp. 742–749 (2003)
6. Hamid, R., Maddi, S., Johnson, A., Bobick, A., Essa, I., Isbell, C.: A novel sequence representation for unsupervised analysis of human activities , a.i journal (2008)
7. Haritaoglu, I., Harwood, D., Davis, L.S.: W4: Real-time surveillance of people and their activities. TPAMI 22(8), 809–830 (2000)
8. Hu, W., Xiao, X., Fu, Z., Xie, D.: A system for learning statistical motion patterns. TPAMI 28(9), 1450–1464 (2006)
9. Johnson, N., Hogg, D.: Learning the distribution of object trajectories for event recognition. In: BMVC '95. pp. 583–592. Surrey, UK, UK (1995)
10. Khalid, S., Naftel, A.: Classifying spatiotemporal object trajectories using unsupervised learning of basis function coefficients. In: VSSN '05: Proc. of Intl Workshop on Video Surveillance & Sensor Networks (2005)
11. Laptev, I., Lindeberg, T.: Space-time interest points. In: ICCV. pp. 432–439 (2003)
12. Morris, B.T., Trivedi, M.M.: Learning and classification of trajectories in dynamic scenes: A general framework for live video analysis. AVSS'08 (2008)
13. Owens, J., Hunter, A.: Application of the self-organizing map to trajectory classification. In: VS '00: Proc. of the Third IEEE Int. Workshop on Visual Surveillance (2000)
14. Piciarelli, C., Foresti, G.L.: On-line trajectory clustering for anomalous events detection. Pattern Recogn. Lett. 27(15), 1835–1842 (2006)
15. Porikli, F.: 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)
16. Romdhane, R., Mulin, E., A. Derreumeaux, N.Z., Piano, J., Lee, L., Leroi, I., Mallea, P., David, R., Thonnat, M., Bremond, F., Robert, P.: Automatic video monitoring system for assessment of alzheimer's disease symptoms. JNHA - The Journal of Nutrition, Health and Aging Ms. No. JNHA-D-11-00004R1 - 2011.
17. S.Calderara, Cucchiara, R., Prati, A.: Detection of abnormal behaviors using a mixture of von mises distributions. In: IEEE AVSS 2007
18. Shi, J., Tomasi, C.: Good features to track. In: IEEE CVPR'94. pp. 593 – 600 (1994)
19. Zouba, N., Bremond, F., Thonnat, M.: Multisensor fusion for monitoring elderly activities at home. In: AVSS'09. Genoa, Italy. (September 2009)