DIRAC
Detection and Identification of Rare Audiovisual Cues

Insperata accident magis saepe quam quae speres.
(Things you do not expect happen more often than
things you do expect) Plautus (ca 200 (B.C.))

Project no: 027787

DIRAC

Detection and Identification of Rare Audio-visual Cues

Integrated Project
IST – Priority 2

DELIVERABLE NO: D3.13
Concepts of Self-learning Tracker Trees

Date of deliverable: 31.12.2010
Actual submission date: 01.02.2011

Start date of project: 01.01.2006 Duration: 60 months

Organization name of lead contractor for this deliverable: ETHZ

Revision [0]

Project co-funded by the European Commission within the Sixth Framework Program (2002-2006)		
Dissemination Level		
PU	Public	
PP	Restricted to other program participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	X

DIRAC

Detection and Identification of Rare Audiovisual Cues

Insperata accident magis saepe quam quae speres.
(Things you do not expect happen more often than
things you do expect) Plautus (ca 200 (B.C.))

D3.13 – CONCEPTS OF SELF-LEARNING TRACKER TREES

EIDGENOESSICHE TECHNISCHE HOCHSCHULE ZUERICH (ETHZ)

Abstract:
We present two different concepts of self-learning tree-like structures that model normal human behavior or activities and are used for the detection of abnormalities. Thanks to the use of hierarchies, we can apply the DIRAC-specific reasoning of novel events. The two approaches have different advantages, they are both applied to the scenario of assisted living, one of the approaches furthermore extends to other surveillance tasks, such as the monitoring of public places, whereas the other one is updatable at runtime.

Note: As long as not all of the described techniques are published, this deliverable remains confidential.

Insperata accident magis saepe quam quae speres.
(Things you do not expect happen more often than things you do expect) Plautus (ca 200 (B.C.))

Table of Content

1. Introduction

1.1. Context

Tracker trees, as they are developed in the DIRAC project and described in previous deliverables (D3.9, D3.12) and publications [Nater *et al.*, 2009], offer a convenient and effective approach to the detection of abnormal events in the context of autonomous living of elderly people. In DIRAC deliverable 3.12, a fully functional implementation of a tracker tree has been described, including the use of different techniques for the various tracker nodes, as well as the quantitative evaluation in the described setting. As the different trackers have been trained specifically to the underlying actions or activities, it is not only possible to reason on incongruencies in the tree, but also detect what specific action the person is performing when he is behaving according to expectations.

However, this supervised tracker tree approach has different shortcomings:

- As the training is done in a supervised setting, where all video frames in the training sequences have to be manually annotated with an action label, establishing a new tracker tree or augmenting the tracker tree with new actions is cumbersome. A more automatic, data driven technique is called for.
- Once the tracker tree is trained, it remains fixed during the time it is applied in people's homes. However, concepts of normality might change over time, furniture might be moved, etc., thus it is desirable to make the approach self-learning during its runtime.
- The tracker tree has proven to perform adequately in the scenario of elderly people surveillance, where one single person has to be monitored in a living room setting. The DIRAC concept of detecting incongruent events would however benefit from showing applications in other scenarios, such as for example the surveillance of public places. This requires the use of different feature types, and again encourages self-learning techniques, as every surveillance setting has its own specificities.

Due to these reasons, we investigated in techniques that automatically build up a hierarchy from the available input image stream. This hierarchy can then be used to analyze unseen data and detect abnormalities. If abnormalities persist, one might reason that the concept of normality has changed, and data that is seen at runtime should be integrated in the model (life-long learning).

1.2. Approaches

We have developed two different approaches for unsupervised analysis of (human) behavior. Both of them learn automatically from training data in order to create effective models, which are applied at runtime. The two proposed approaches are scientifically disseminated, the first one, entitled *Exploiting simple hierarchies for human behavior analysis* was published at CVPR 2010, the second one, with the title *Ad seriatim: temporal relations in video for unsupervised activity analysis* is in the revision process for CVPR 2011.

In this deliverable, we give a very brief description of each of the approaches, and show their implications in the DIRAC project. For details on the technical backgrounds, implementation and experimental evaluation, we refer to the papers that are provided in the appendix.

2. Exploiting simple hierarchies for human behavior analysis

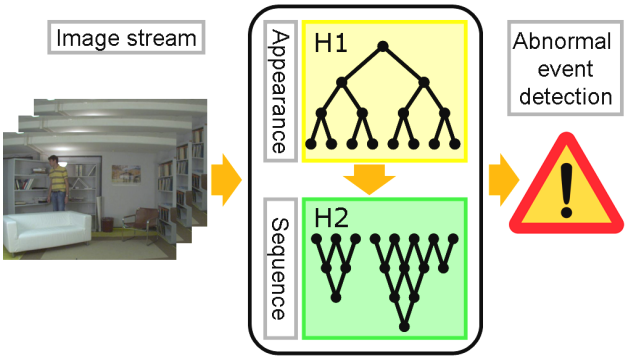


Figure 1. The input image stream is analyzed in a model, which consists of two cascaded hierarchies. They are established in an unsupervised manner. Reasoning, in line with the DIRAC-specific theory is applied in order to perform abnormal event detection in unseen videos.

2.1. Idea

Inspired by neurophysiological findings that suggest that the human visual system makes a distinction between the analysis of instantaneous snapshots (or single poses in the context of human behavior) and the analysis of motion (or actions) [Lange and Lappe, 2006], we set up a two-stage model. It is learned in an unsupervised manner and structures the underlying data in two cascaded hierarchical representations. This is illustrated in Fig. 1. The first stage encodes human appearances on a per-frame basis (snapshot) and is built in a top down process. The second hierarchy explains sequences of appearances, *i.e.*, actions or behavioral patterns, in a bottom up analysis. The overall approach is based on the assumption that normal activities are observed frequently, whereas very rare observations tend to be abnormal. Thus, only frequently occurring appearances and actions are modeled.

At runtime, the model can be used for the interpretation of unseen video data. The data is analyzed with respect to snapshots (single frames) in the first hierarchy as well as sequences that present motion patterns. Abnormal events are detected, based on the DIRAC-specific paradigm of classifier disagreement. Additionally, we show in the paper how to update both hierarchies. In fact, as we apply simple modeling and clustering techniques, an adaptation of the models is feasible. Similar outliers that occur frequently are assumed to belong to a drifted concept of normality, and are therefore added to the initial model.

The comparison between the monkeys' behavioral responses to video stimuli of human walkers and the output of the model, trained on the same human walkers, is investigated in a different DIRAC collaboration and reported in [Nater *et al.*, 2010].

2.2. Relations to DIRAC

A key ingredient of the proposed technique is the hierarchical structure, which is used for modeling. We explicitly seek to create hierarchies that consist of many layers, by fixing the splitting of each tracker node into two sub-nodes. Each sub-node is then more specific than its parent, since it only models a part of the data that the parent node had included. In DIRAC terms, we deal with a *disjunctive hierarchy*.

Abnormal events are detected where and when a more general node in the hierarchy still explains the data, but none of the more specific connected children nodes does. Such an abnormal event can occur at any level in the hierarchy. Since more subtle abnormal events will be outliers at levels more towards the leaf nodes, while important deviations tend to be detected already close to the root, this hierarchy

paves the way for a semantic interpretation on the nature of the detected abnormality.

3. Temporal relations in video for unsupervised activity analysis

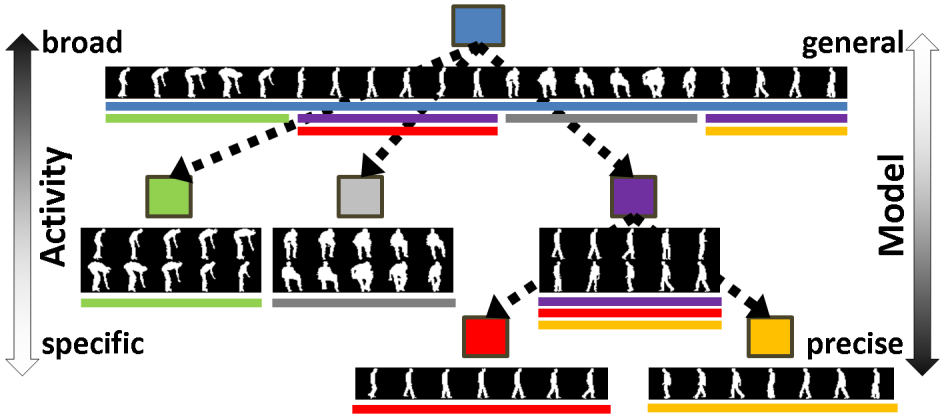


Figure 2. Overview of the proposed hierarchical model. Each node represents part of the data. Nodes at higher levels are more general and describe broad activity data in an approximate manner, nodes on lower levels are well tuned to specific activities. In order to successively split the data, temporal relations in the image stream are exploited.

3.1. Idea

Temporal consistency is an important cue in continuous data streams such as videos. By exploiting the temporal relations from frame to frame, we show that it is possible to split the data stream into its composing activities. This splitting procedure is applied recursively on the data, as sketched in Fig. 2, using discriminative slow feature analysis (SFA). This technique, first introduced in [Wiskott and Sejnowski, 2002] is inspired by human learning capacities. Once all the activities are extracted from the data, a simple PCA-based modeling technique is applied to be able to interpret unseen data at runtime. As shown in the experimental section of the appended (submitted) paper, the technique yields superior results compared to the approach described in Sec. 2 on the same in-house data. Furthermore, it can be applied to different surveillance scenarios by interchanging feature types. We show

results for traffic monitoring as well as the surveillance of a public place from webcam data.

3.2. Relations to DIRAC

As in the previous approach, the concept that enables the detection of abnormal events is the reasoning in the hierarchy. Following DIRAC terminology, we establish a *disjunctive hierarchy*, where nodes closer to the root are more general and become more and more specific when moving towards the leafs. The splitting technique is now more data-driven than previously, since the number of established sub-nodes depends on the low dimensional distribution of data points in SFA space.

Abnormal events are again detected where and when a more general node can well describe the current observation, but none of its more specific children nodes can. From the location in the tree, where this abnormal event occurs, reasoning on the severity of the event can be deduced. Due to the SFA modeling, nodes in this model often precisely correspond to activities or actions with a semantic meaning and thus a semantic interpretation of the abnormal event is encouraged.

4. Conclusion

The two appended publications present two approaches to the unsupervised modelling of human behaviour and activities in general. In this deliverable, we briefly introduced the techniques, and explained the DIRAC-related considerations. We show that by means of unsupervised methods, the detection of abnormal events in assisted living scenarios is possible.

In an automatic manner, both approaches establish disjunctive hierarchies, which can be used for the interpretation of unseen data at runtime. Whereas the first approach has the advantage of being updatable, the second technique has superior performance and delivers a more easily interpretable hierarchy of activities. In addition, the second concept was also successfully applied to other surveillance scenarios with different feature types.

Unsupervised techniques, compared to the supervised tracker tree come with one particular drawback that is the lack of an exact semantic meaning of each modeled

activity node. While one tracker in the supervised tracker tree exactly corresponded to a well-defined human action, such as walking or sitting, this is less clear when an unsupervised model is built. Therefore the detection and recognition of known actions is less straight forward, and additional human effort would be required to obtain action recognition capabilities as in the supervised tracker tree.

5. References

[Lange and Lappe, 2006] J. Lange and M. Lappe, "A model of biological motion perception from configural form cues", *Journal of Neurosciences*, 26:2894-2906, 2006.

[Nater *et al.*, 2010] F. Nater*, J. Vangeneugden*, H. Grabner, L. Van Gool and R. Vogels, "Discrimination of locomotion direction at different speeds: A comparison between macaque monkeys and algorithms", *ECML Workshop on rare audio-visual cues*, 2010. (*equally contributing first authors)

[Nater *et al.*, 2009] F. Nater, H. Grabner, T. Jaeggli and L. Van Gool, "Tracker trees for unusual event detection", *ICCV Workshop on Visual Surveillance*, 2009.

[Wiskott and Sejnowski, 2002] L. Wiskott and T. Sejnowski, "Unsupervised learning of invariances" *Neural Computation*, 14(4):715-770, 2002.

More references can be found in the reference sections of the appended papers.

6. Appendix

Exploiting Simple Hierarchies for Unsupervised Human Behavior Analysis

Fabian Nater[1] Helmut Grabner[1] Luc Van Gool[1,2]

[1]Computer Vision Laboratory
ETH Zurich
{fnater,grabner,vangool}@vision.ee.ethz.ch

[2]ESAT - PSI / IBBT
K.U. Leuven
luc.vangool@esat.kuleuven.be

Abstract

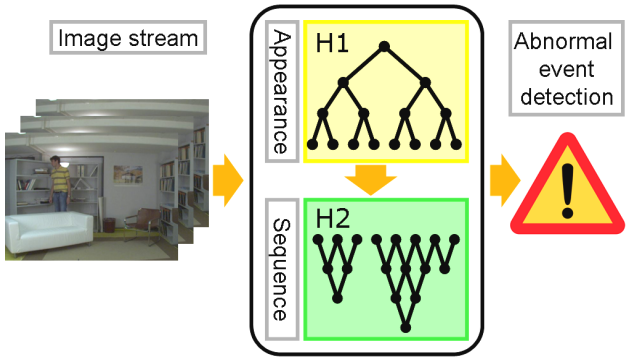
We propose a data-driven, hierarchical approach for the analysis of human actions in visual scenes. In particular, we focus on the task of in-house assisted living. In such scenarios the environment and the setting may vary considerably which limits the performance of methods with pre-trained models. Therefore our model of normality is established in a completely unsupervised manner and is updated automatically for scene-specific adaptation. The hierarchical representation on both an appearance and an action level paves the way for semantic interpretation. Furthermore we show that the model is suitable for coupled tracking and abnormality detection on different hierarchical stages. As the experiments show, our approach, simple yet effective, yields stable results, e.g. *the detection of a fall, without any human interaction.*

1. Introduction

In many visual surveillance scenarios, an automatic system has to detect anomalies and then give out a warning for an operator. To cope with various situations and environments, a multitude of different approaches have been proposed, see [8] for a survey. Most of these methods detect anomalies as outliers to previously trained models of normality. Successes include the analysis of an agent's motion patterns [19], traffic monitoring [10], the surveillance of public places [1], and the evaluation of a webcam image stream [6].

Our work aims at supporting autonomous living of elderly or handicapped people, by monitoring their well-being with a visual surveillance system installed in their homes. Fall detection is one major task of such activity monitoring systems [18]. To this end, rule-based systems have been established, performing well for the detection of different, predefined dangerous cases (*e.g.* [2, 15]). They lack general applicability, however. Other methods implement a more principled model of human behavior and are then able to point out suspicious configurations. Boiman and Irani [4], for example, check whether they can explain a given activity as a puzzle of spatio-temporal database entries. In our previous work [17], we used a set of trackers, each dedicated to a certain range of activities. Another approach is to extract key-frames and estimate transition probabilities for a set of predefined activities (*e.g.* [14]). The main limitation of all these systems is their need for an offline, prior training with labeled training data. In such supervised approaches, no long-term adaptation to particular scenes or persons can be achieved. Furthermore, no training sequence contains a comprehensive set of all the situations to expect. Due to these reasons, a dynamical, data-driven

(a) Normal action (b) Abnormal event



(c) Human behavior analysis

Figure 1. Human behavior in an input image stream is analyzed in a cascade of two hierarchical models. They are established in an unsupervised manner and permit the characterization of normal and abnormal events for example in in-house monitoring scenes.

model is called for. In a more unsupervised setting, recent work [7] uses very weakly annotated image sequences in order to learn actions autonomously.

In this paper we propose to learn a model of normal human behavior in a completely unsupervised manner. This model consists of two hierarchical representations arranged in a cascade, as illustrated in Fig. 1(c). The first stage encodes human appearances and is built by a top-down process, whereas the second hierarchy explains sequences of appearances (*i.e*. actions or behavioral patterns) and is built by a bottom-up analysis. In fact, given a sequence of images, we first map these images to a finite set of symbols describing *what* is observed. Secondly, we analyze the sequence of symbols to characterize in *which order* the observations occur. We call these sequences *micro-actions* since they usually correspond to basic body motions. Finally, the evaluation could be augmented by learning the temporal (*e.g*. within a day or a week) and spatial dependencies. All this together models the normal behavior of a person in a scene. At runtime, this structure is used as a model of normality to which unseen data is compared. The person is tracked and statistical outliers with respect to appearance and action are detected robustly at different hierarchical levels. We additionally show how to update this model in order to incorporate newly observed normal instances.

The paper is organized as follows: In Sec. 2 and Sec. 3 we introduce the hierarchical representations for appearances and actions, respectively. Sec. 4 shows the target tracking and abnormal event detection on unseen data, Sec. 5 discusses the model update procedure. Experiments are presented in Sec. 6 and the paper is concluded in Sec. 7.

2. Appearance hierarchy ($H1$)

We start from an image stream

$$\mathcal{S} = \langle \mathbf{x}_1, \ldots, \mathbf{x}_T \rangle, \quad \mathbf{x}_t \in \mathcal{X} \tag{1}$$

of T frames which is described in an arbitrary feature space $\mathcal{X}$. The goal is to group similar image descriptors together and create a finite number of clusters representing the data in a compact form. Hence, we propose to use a k-means clustering algorithm [11], applied hierarchically to the training data in a top-down procedure with a distance measure $d(\mathbf{x}_i, \mathbf{x}_j)$ defined in $\mathcal{X}$. The root node cluster $\mathcal{C}^{(1)}$ describes all $\mathbf{x}_t \in \mathcal{S}$. Moving down in the hierarchy, the data associated to one cluster on layer l, *i.e*. $\mathcal{C}^{(l)} \subseteq \mathcal{X}$ is separated into k sub-clusters on layer $l+1$. This process is repeated until a certain stopping criterion is met, for example when the number of data points in a cluster gets too small. An example of the resulting tree structure $H1$ is presented in Fig. 2 using $k = 2$.

By creating a hierarchical representation, the clusters become more specific when moving down the tree structure.

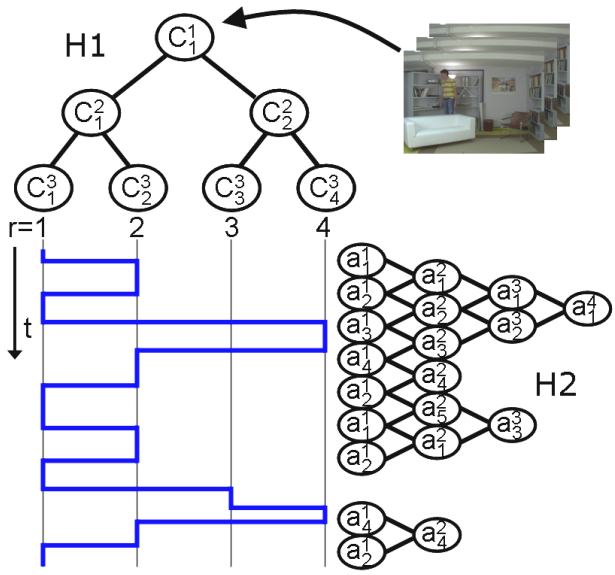


Figure 2. Illustration of the unsupervised learning approach, composed of two hierarchies. In $H1$, a sequence of images is mapped by clustering to a number of discrete symbols, in $H2$ the sequence of these symbols is analyzed.

While the cluster at the root node has to describe all $\mathbf{x}_t$ in the training set and thus exhibits a large intra-cluster variance, clusters at lower layers only contain similar data and therefore describe this data more precisely.

Eventually, each feature vector $\mathbf{x}_t$ is mapped to a symbol r_t which is the number of its corresponding leaf node cluster. The image stream is accordingly expressed by the sequence of symbols, *i.e*.

$$\mathcal{S} \mapsto \mathcal{R} = \langle r_1, \ldots, r_T \rangle, \quad r_t \in \mathbb{N} \cup \{\sharp\}. \tag{2}$$

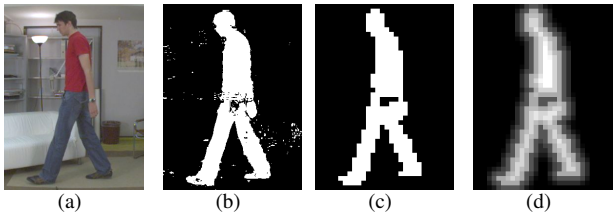
In order to obtain compact clusters, we remove statistical outliers at every clustering step with the formulation of Sec. 4.1. The symbol $r = \sharp$ is assigned to an $\mathbf{x}_t$ that is not matched to a leaf node cluster. For their use at runtime, all obtained clusters $\mathcal{C}_i^{(l)}$ are represented with their centers $\mathbf{c}_i$ and the distribution $D_i^{(l)}$ of distances $d_i = d(\mathbf{c}_i, \mathbf{x})$ of all the samples $\mathbf{x}$ assigned to this cluster.

Illustration

We demonstrate the mapping of input images to clusters in the tree structure. An indoor training sequence[1] of about $7,100$ images was recorded at 15 frames per second in VGA resolution. It contains diverse 'every-day' actions such as walking, walking behind occluding objects, sitting on different chairs, picking up small objects, *etc*., repeated a few times.

[1]Data available from `www.vision.ee.ethz.ch/fnater/`.

Feature extraction. We apply background subtraction[2] on the input images for the extraction of foreground blobs. The resulting silhouettes are rescaled to a fixed number of pixels (40×40 in our case) and a signed distance transform is applied. Maximum and minimum pixel values are bounded and an offset is added to obtain non-negative values (*c.f.* Fig. 3). Finally, the rows are concatenated in a vector that defines the fixed length image features $\mathbf{x}$ ($N = 1600$), describing the appearance of one person in the scene.



(a) (b) (c) (d)

Figure 3. Feature extraction: (a) original, (b) segmented, (c) post-processed and rescaled, (d) distance transformed.

Distance measure. As a distance measure to compare the feature vectors in the clustering procedure, we use the χ^2 test statistic as in [3]. Two samples $\mathbf{x}_u$ and $\mathbf{x}_v$ with elements $x_u(n)$ and $x_v(n)$, $n = 1 \dots N$ are at a distance

$$d(\mathbf{x}_u, \mathbf{x}_v) = \frac{1}{2} \sum_{n=1}^{N} \frac{[x_u(n) - x_v(n)]^2}{x_u(n) + x_v(n)}. \qquad (3)$$

This said, the silhouette features are extracted and clustered ($k = 2$) in order to build $H1$. The outcome is visualized in Fig. 4, where a random set of silhouettes is shown for each cluster at different layers. Similar appearances are grouped well into the same cluster for a hierarchcal depth of $l = 5$ already.

3. Action hierarchy ($H2$)

As depicted in Fig. 2, we start from the sequence of symbols $\mathcal{R}$ defined in Eq. (2). The goal is to exploit the information in this sequence and extract frequent patterns which we refer to as micro-actions. Their variable length naturally defines a hierarchy, since longer actions automatically represent more information. Our approach is inspired by the work of Fidler *et al*. [9], where neighboring generic visual parts are combined in a hierarchy, in order to form entire objects on higher levels. At each level only the statistically relevant parts are chosen in order to omit noise. Since our input is a one-dimensional state sequence, we combine temporally adjacent generic parts (micro-actions) for the hierarchical combination of new, more informative ones.

[2]We operate on static camera images and in scenes with few moving objects, but other appearance features could be used similarly. However, we did not notice any failures of our approach that were caused by bad foreground segmentation.

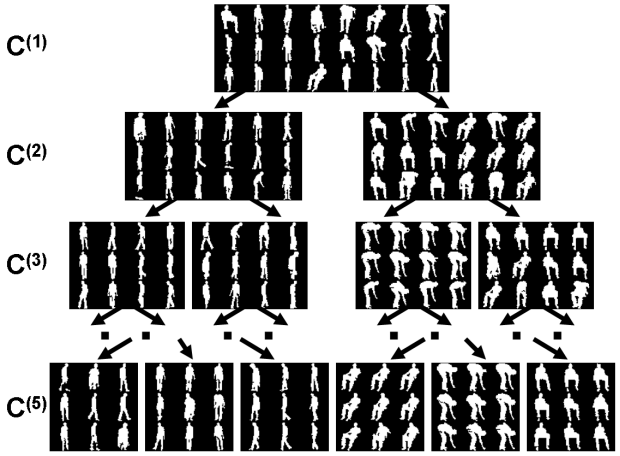


Figure 4. Visualization of the proposed binary tree for the hierarchical appearance representation ($H1$). For each of the displayed clusters at different layers $C_i^{(l)}$, randomly chosen silhouettes are displayed.

More in detail, we first define a set of basic actions $a_i^{(1)}$ that encode a state change $r_t \rightarrow r_{t+1}$ in the sequence of symbols:

$$\mathcal{A}^{(1)} = \{a_i^{(1)} := r_t \rightarrow r_{t+1} \mid r_t \neq r_{t+1}, P(a_i^{(1)}) > \theta_{act}\}, \qquad (4)$$

where $P(a_i)$ is the occurrence probability of the micro-action a_i. The parameter θ_{act} is defined such that only frequently occurring symbol changes are considered, thereby discarding spurious changes. From the second level on, higher level micro-actions with length λ are the combination of lower level micro-actions, *i.e.*

$$\mathcal{A}^{(\lambda)} = \{a_i^{(\lambda)} := a_p^{(\lambda-1)} \rightarrow a_q^{(\lambda-1)} \mid P(a_i^{(\lambda)}) > \theta_{act}\}. \qquad (5)$$

The frequency condition θ_{act} naturally introduces a limit on the maximal length of the micro-actions (longer micro-actions appear less frequently). The symbol $r = \sharp$, attributed to a feature vector which is not matched to any leaf node cluster, is excluded from the description of any a_i^{λ}.

We want to be independent of a labeling of the states (they might even not be attributed a clear label as they are learned through an unsupervised procedure) and the method we propose relies much more on the assumption that, within the target scenario, normal actions are likely to be repeated. This fact is exploited for the extraction of usual temporal patterns. Summarizing, we continuously replace the original sequence of symbols $\langle r_1, \dots, r_t \rangle$ by frequent patterns a_i^{λ} and we can represent the image stream as a series of *micro-actions* of different lengths λ:

$$\mathcal{S} \mapsto \mathcal{R} \mapsto \langle a_1^{(\lambda)}, \dots, a_t^{(\lambda)} \rangle, \quad a_i^{(\lambda)} \in \mathcal{A}^{(\lambda)}. \qquad (6)$$

Note that in this formulation, micro-actions can overlap, which is in line with the observation that often no clear-cut

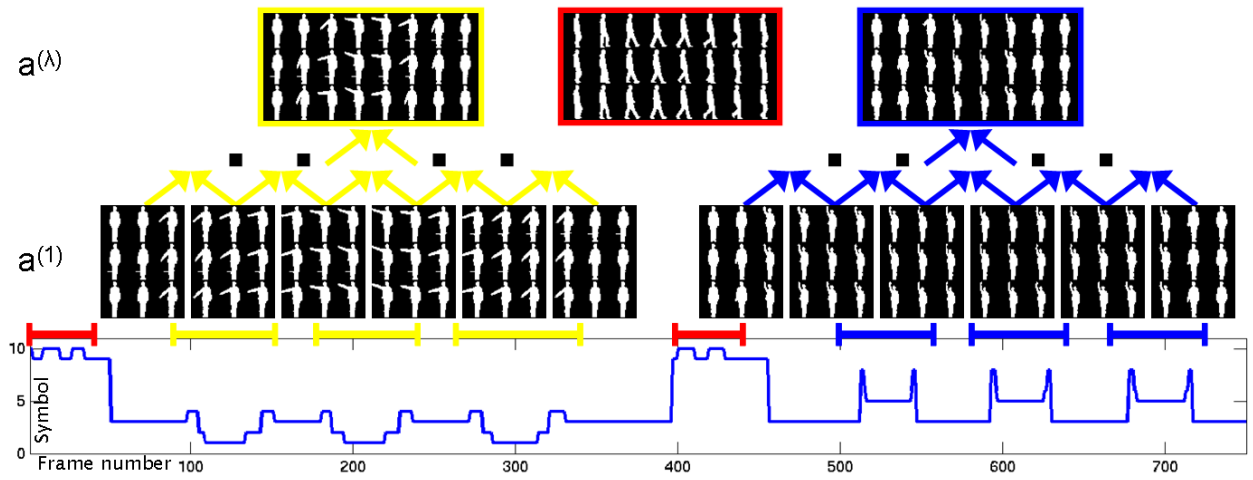
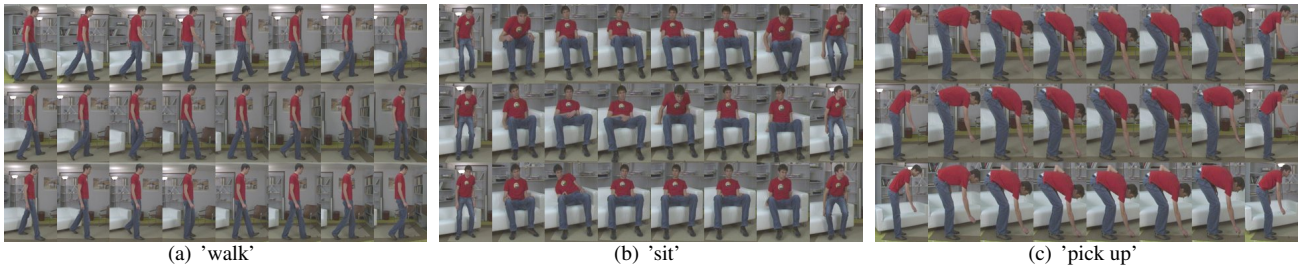


Figure 5. Illustration of the micro-action hierarchy ($H2$) for the action recognition test dataset [13]: Micro-actions are extracted from symbol transitions and can be combined gradually into higher level actions.



(a) 'walk' (b) 'sit' (c) 'pick up'

Figure 6. Examples of segmented actions as produced with our method. In an unsupervised manner repetitive microactions are extracted, which can be labeled manually, if desired. Repetitions in the training dataset are presented in rows.

boundaries of actions can be defined [16].

Illustration

Action recognition We employ a publicly available action recognition dataset [13] to illustrate the extraction of micro-actions and select two right arm motions ('turn left' and 'stop left'). The two sequences additionally have introductory walking, they are sticked together and analyzed as shown in Fig. 5. Binary silhouettes are provided in the dataset. The plotted sequence of symbols is obtained with the procedure of Sec. 2. In a next step, repeated patterns in this sequence are extracted first on the basic level $a^{(1)}$ (*i.e.* transitions, Eq. (5)), then growing in length on higher levels (Eq. (6)). The finally meaningful micro-actions are presented in the upper part of Fig. 5 and correspond to the actions to be recognized.

Indoor surveillance If we apply the same procedure to the previously described indoor training video, the sequence of symbols is more complex and various repeated micro-actions appear at different hierarchical levels. A selection is shown in Fig. 6. In case the system would be required to constantly report activities, they could be labeled manually for ease of human reference ('walking', 'sitting down', 'getting up', 'picking up from the floor'). This split into units that intuitively correspond to basic actions, demonstrates that within the repeated action context, it is possible to isolate and segment these actions in an unsupervised manner.

4. Tracking and abnormality detection

In this section, we show how the established model of normality is employed for the runtime analysis of unseen images. $H1$ will be used for *tracking* and the interpretation of the *appearance*, $H2$ is used for the interpretation of *actions*. In both hierarchies, abnormalities can be spotted.

4.1. Data-dependent inlier

Given a query image with extracted features $\mathbf{x}$, we want to determine its cluster membership $\mathcal{C}_i$ based on the distance $d(\mathbf{x}, \mathbf{c}_i)$. According to the *curse of dimensionality*, distances in high dimensional spaces tend to lose their significance and it is therefore difficult to find a fixed distance threshold for the classification of the query. Hence, we apply the concept of data-dependent inlier [12], comparing $d(\mathbf{x}, \mathbf{c}_i)$ to the

distance distribution D_i of the cluster $\mathcal{C}_i$. The probability that the query point $\mathbf{x}$ is an inlier to $\mathcal{C}_i$ is

$$p_{inlier}\left(d(\mathbf{x}, \mathbf{c}_i)\right) = 1 - \int_{\xi=0}^{d(\mathbf{x},\mathbf{c}_i)} D_i(\xi) d\xi. \qquad (7)$$

For classifying a sample as inlier, its inlier probability must exceed a certain threshold:

$$p_{inlier}\left(d(\mathbf{x}, \mathbf{c}_i)\right) \geq \theta_{inlier}. \qquad (8)$$

In the analysis of unseen data, we keep $\theta_{inlier} = 0.05$ which means that $\mathbf{x}$ is classified as outlier if its distance to the considered cluster center is larger than 95% of the data in that cluster.

4.2. Tracking

In every frame we want to determine the location and scale of the bounding box (*i.e.* find $\mathbf{x}_t$) that best matches the trained model. This is important for a stable symbol mapping as well as a precise tracking of the human target. We apply a best search strategy in which the local neighborhood of the output at the previous time step is exhaustively scanned. Each feature representation $\mathbf{x}'_t$ extracted from a hypothesized location and scale is evaluated by using Eq. (8) and is propagated as far as possible in $H1$. With this formulation, an $\mathbf{x}'_t$ can sometimes be matched to more than one cluster on the same layer. In that case, all connected lower layer clusters are evaluated subsequently. As tracking result $\mathbf{x}_t$, the hypothesis which applies to a cluster at the lowest possible layer with maximal p_{inlier} is searched for. Ideally this is a leaf node cluster and its symbol r_t is attributed to $\mathbf{x}_t$. If no leaf node cluster is reached, no symbol can be attached to this observation. Furthermore, if the observation is already outlier to the root node cluster, the target cannot be tracked in H1. In order not to lose the target, we simultaneously run a mode estimating tracker [5], which specifies the output in this case. In our current implementation, this tracker is also used to establish a prior for the exhaustive search, which additionally speeds up the procedure.

4.3. Abnormal appearance

An abnormal (or novel) appearance is identified in $H1$ on hierarchical level l if the tracking result $\mathbf{x}_t$ is inlier to at least one cluster at level l but is outlier to all of its connected clusters in layer $l+1$. Since no leaf node can be matched to $\mathbf{x}_t$ in this case, the symbol $r_t = \sharp$ is attributed, characterizing an unknown (not matching) state. Of course, if $\mathbf{x}_t$ is outlier at the root node already, it is also abnormal. Although the tree-like model is learned in an unsupervised manner, it helps to order and interpret anomalies. Completely new poses tend to be outliers to clusters close to the tree root already, while not that different poses are matched on some layers before being detected as outliers. Hence, and as we will show in the experimental section, this hierarchy assists with a semantic interpretation of the abnormal poses.

4.4. Abnormal actions

Abnormal action analysis is based on the mapping $\mathcal{S} \mapsto \mathcal{R}$ and the hierarchical model of usual actions encoded in the hierarchy $H2$. In that sense, the sequence $\mathcal{R}$ is scanned for its correspondence to $\mathcal{A}^{(\lambda)}$.

The sequence of symbols r_t extracted at runtime is analyzed as in Eq. (4) and Eq. (5) and combined into micro-actions $a_i^{(\lambda)}$ with different lengths λ. Each micro-action is then compared to the set of normal micro-actions $\mathcal{A}^{(\lambda)}$. If it is found in the database, it is considered to be normal behavior at level λ. The length of the action is used to know how usual the behavior is. If $\mathbf{x}_t$ is mapped to the unknown state $r_t = \sharp$, no micro-action can be established and the sequence analysis breaks down temporarily.

4.5. Scene context

Additionally, our approach can be embedded in a scene context learning framework. There are a certain number of events or actions which can be usual in one part of the scene but are not in another one. Thinking of in-house visual surveillance, this might be the presence of a person lying on a couch *vs*. the person lying on the floor. Considering only human appearances, the two scenarios might look the same, but with additional scene information, they could be told apart. Then, the second case could be pointed out as abnormal.The same idea applies to actions performed at a certain time of day, *e.g*. a person observed walking through a living room at 4 a.m. should not necessarily be considered normal. However, these techniques lie beyond the scope of this paper.

5. Update procedure

After the training phase, the model of normal behavior usually remains fixed. Obviously, not all possible appearances and actions can be learnt off-line, due to the lack of sufficient training data. Furthermore, the *normality concept* might change over time and thus the model needs to be adapted continuously. For example, a different walking style like limping is (correctly) classified as abnormal since it can not be modeled through a normal action sequence. Yet, if it starts to appear frequently, it might turn into a normal behavior, *e.g*. due to a lasting deterioration of the person's physical state. It is therefore desirable to design a dynamic method, able to extend (or even shrink) the model of normality.

5.1. Appearance update

The hierarchical model $H1$, can essentially be modified in two ways. Firstly, new appearances which are classified as outliers at runtime might need to be included if they occur often. Secondly, some existing cluster could be further refined, *e.g*. for the distinction between two persons. Since we focus on the scenario where a single person should be monitored when left to his own devices, we will only deal with the first case as yet. It is clear that for long-term, real-world usage, the system should be enriched with a method to identify the person of interest and to notice the presence of others (like care-takers).

At runtime we collect all feature vectors that are outliers at a certain layer in the hierarchy. During a supporting phase (*e.g*. when the system is in an idle mode since no person is in the room) we incrementally update the hierarchy. The creation of new clusters is investigated at the specified layer, besides the existing ones. To that end, we apply the same hierarchical clustering approach to the set of outliers. It is important not to change the existing hierarchy since already established knowledge should not get lost. Assuming that also 'real outliers' could be in the update data, we follow a restrictive policy and set the threshold θ_{inlier} (Eq. (8)) to a high value already for clustering. Finally, new leaf node clusters are established and new symbols are defined.

5.2. Micro-action update

Established micro-actions by definition have a sufficient frequency of occurrence (Eq. (5)). We propose to estimate these probabilities incrementally, by updating them with new observations using the principle of exponential forgetting. Hence, frequent, new micro-actions become available for the next level and less frequent micro-actions are removed. Micro-actions using new symbols in $H1$ are included automatically, since they will first get picked up by lower levels (Eq. (4)) and then might be used for longer micro-actions as soon as they occur often.

Summarizing, one could start with an empty database, with everything considered abnormal at the beginning. When humans (moving objects in general) are observed several times, first appearances and later micro-actions are added to the model of normality.

6. Experiments

In this section, we validate the proposed approach with a series of experiments. To the best of our knowledge, there is no standard dataset for testing in-the-home visual monitoring techniques. As the experiments will show, the method is successful at detecting salient appearances and behaviors also from a human point of view. We want to re-emphasize at this point, that the main goal of this work is to assist in the prolonged, independent living of elderly or handicapped people. Hence, we focus on scenarios with only that single person in the scene. As such system would need to be deployed in many homes, the unsupervised approach behind it is of particular importance.

6.1. Behavior analysis

The test footage of about $1,000$ images was recorded in the same setting as the training sequence (*c.f*. Sec 2), but now contains abnormalities such as heavy waving, jumping over the sofa and a fall. The model of normality was established as explained previously (appearance clustering in Fig. 4, extraction of frequent micro-actions like the ones in Fig. 6), and we now want to explain the test sequence by means of this model. The target person is tracked and appearances and actions are interpreted. A selection of the per-frame results are visualized in the top part of Fig. 7.

The color of the bounding box indicates the layer l in $H1$ farthest from the root, on which the observation is still considered normal according to Eq. (8). A red bounding box is drawn if the observation is outlier to the root node, (its dimensions are in that case determined by the mode estimating tracker [5]), nuances of orange are used for intermediate layers and green encodes an appearace that is described in a leaf node.

The vertical black bar on the left side of the bounding box represents the level λ in $H2$ on which the sequence of symbols is normal. The bar is resized accordingly. In case the appearance does not reach a leaf node in $H1$, *i.e*. the bounding box is not green, the action level cannot be calculated and therefore vanishes.

The plots on the bottom part of Fig. 7 indicate three temporal characteristics: (i) The maximal inlier probability (in the matching cluster) remains at high value and is stable as long as one leaf node cluster is matched. We also show the 5% threshold θ_{inlier} which is used for the classification of abnormalities. (ii) The matching cluster identity (symbol r_t) changes over time ($0 = \sharp$) which allows for the recognition of (iii) micro-actions. They are matched hierarchically and the maximal length is visualized. Two patterns ('walking' and 'sitting') are highlighted which in fact correspond to the same micro-actions as shown in Fig. 6(a) and Fig. 6(b).

We now run through a number of interesting episodes in the test video. In (a) everything is normal, the action level is not so high yet since the sequence just started. (b) and (i) are two abnormal events at different levels within $H1$, whereas (e), (g) and (h) are outliers to the root node already. In these cases, a practical system would probably generate an alarm. Note that lying on the couch (g) was not present in the training set, therefore it is judged abnormal at first. On the other hand, occlusions were trained for and their handling in (d) does not cause problems. It is interesting to compare (c) and (f): Although the same appearances are

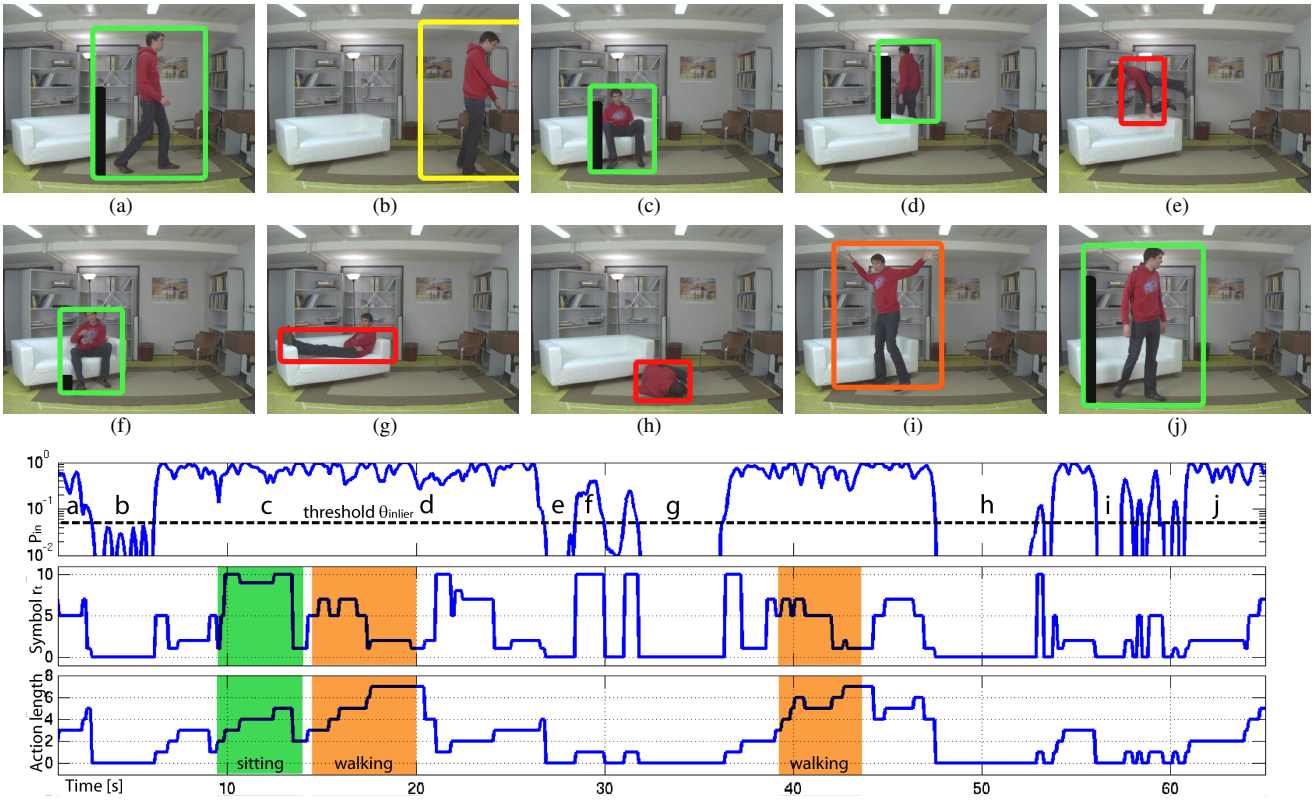


Figure 7. Our method tracks the person, analyzes the appearance in $H1$ and interprets the micro-action in $H2$. Top: Various normal and abnormal instances of the test sequence are presented. The color of the bounding box encodes the layer in $H1$, on which the observation is normal, the length of the black bar on the left side of the bounding box indicates the micro-action level. Bottom: Three representative values are plotted over time, the inlier probability at the leaf node level of $H1$, the matched symbol r_t and the micro-action length $a^{(\lambda)}$. Two actions are highlighted (see text for details, figure is best viewed in color).

present, (f) needs special attention, since it resulted from an unknown action (jumping over the couch in (e)) and hence holds a small black action bar.

6.2. Model update

A second experiment illustrates the benefit of the model update. The video used for the update contains the repeated 'abnormality' of the person lying on the couch but also a real irregular event (*i.e.* the person falls). This set of appearance feature vectors, outliers to the root node of $H1$, is stored during the analysis of the sequence and a randomly chosen sample is presented in Fig. 8(a). All abnormal appearances are used for updating the model though.

After this update, when analyzing yet another video sequence, previously normal appearances stay normal (Fig. 8(b)), lying is now included in the model of normality and handled accordingly (c), while other events remain outliers (d). The model would need to see some more occurrences of lying on the couch in order to also recognize the micro-action 'lying down' as normal. This had not happened yet, whence the small black action bar in (Fig. 8(c)).

For a more precise analysis of the experiments, we manually annotated abnormal events per frame for the sequence of Fig. 7. A RPC plot, depicted in Fig. 9 quantifies the performance by sweeping parameter θ_{inlier} (Eq. (8)). By moving down in $H1$ (from layer 1 to layer 5), a higher precision is achieved, which is essential for our task. At a precision of 98%, the recall increases from 32% (root node level) to

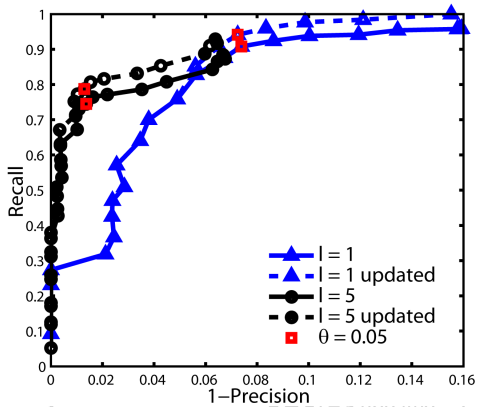


Figure 9. Recall-precision curves for the video sequence of Fig. 7 verifies the applicability of our technique.

(a) (b) (c) (d)

Figure 8. Illustration of the update procedure: (a) some feature vectors and their according image regions taken for the update of $H1$, (b) normal appearances stay normal after the update, (c) lying turned normal after the update and (d) real outliers are still detected.

77% (leaf node level), respectively to 81% after the update. These numbers show both, (i) the effect of using the hierarchical structure $H1$ and (ii) the benefit of updating.

7. Conclusion

We have presented an approach for the unsupervised analysis of human action scenes. In particular, we have focused on an application to support prolonged independent living. The ideas are very general however, and can be extended to other scenarios. The method involves two automatically generated and updated hierarchies learned in an unsupervised manner. One deals with the normal appearances, and from appearance transitions, the second builds up a database of normal actions or episodes. Due to the hierarchical nature of this model of normality, it is easier to name deviations from normality and to analyze those at different semantic levels (a human would still have to give such names to different cases, but that is a small effort). The system is able to adapt itself and can include new modes of normality. Hence, also the semantic level increases and after sufficiently long learning periods, it would become possible to detect deviations from certain routines. Thus, one strategy allows for the detection of abnormal events at different levels of sophistication (e.g. falling or walking with an abnormal gait).

Acknowledgements This work is funded by the EU Integrated Project DIRAC (IST-027787). We further thank Thomas Mauthner for inspiring discussions.

References

[1] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz. Robust real-time unusual event detection using multiple fixed-location monitors. *PAMI*, 30(3):555–560, 2008.

[2] D. Anderson, R. Luke, J. Keller, M. Skubic, M. Rantz, and M. Aud. Linguistic summarization of video for fall detection using voxel person and fuzzy logic. *CVIU*, 113(1):80–89, 2009.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(4):509–522, April 2002.

[4] O. Boiman and M. Irani. Detecting irregularities in images and in video. In *Proc. ICCV*, 2005.

[5] G. R. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, (Q2), 1998.

[6] M. D. Breitenstein, H. Grabner, and L. Van Gool. Hunting nessie – real-time abnormality detection from webcams. In *ICCV WS on Visual Surveillance*, 2009.

[7] P. Buehler, M. Everingham, and A. Zisserman. Learning sign language by watching tv (using weakly aligned subtitles). In *Proc. CVPR*, 2009.

[8] H. M. Dee and S. A. Velastin. How close are we to solving the problem of automated visual surveillance? *Machine Vision and Applications*, 19(5-6):329–343, 2008.

[9] S. Fidler, G. Berginc, and A. Leonardis. Hierarchical statistical learning of generic parts of object structure. In *Proc. CVPR*, 2006.

[10] W. Hu, X. Xiao, Z. Fu, D. Xie, F.-T. Tan, and S. Maybank. A system for learning statistical motion patterns. *PAMI*, 28(9):1450–1464, 2006.

[11] A. Jain, M. Murty, and P. Flynn. Data clustering: A review. *ACM Computer Surveys*, 31(3):264–323, 1999.

[12] E. M. Knorr and R. T. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proc. Int. Conf. on Very Large Data Bases*, 1998.

[13] Z. Lin, Z. Jiang, and L. S. Davis. Recognizing actions by shape-motion prototype trees. In *Proc. ICCV*, 2009.

[14] F. Lv and R. Nevatia. Single view human action recognition using key pose matching and viterbi path searching. In *Proc. CVPR*, 2007.

[15] A. Nasution and S. Emmanuel. Intelligent video surveillance for monitoring elderly in home environments. In *IEEE Workshop on Multimedia Signal Processing*, 2007.

[16] P. Natarajan and R. Nevatia. Online, real-time tracking and recognition of human actions. In *IEEE Workshop on Motion and Video Computing*, 2008.

[17] F. Nater, H. Grabner, T. Jaeggli, and L. Van Gool. Tracker trees for unusual event detection. In *ICCV WS on Visual Surveillance*, 2009.

[18] N. Noury, A. Fleury, P. Rumeau, A. Bourke, G. Laighin, V. Rialle, and J. Lundy. Fall detection - principles and methods. In *IEEE Engineering in Medicine and Biology Society*, 2007.

[19] C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *PAMI*, 22(8):747–757, 2000.

Ad Seriatim: Temporal Relations in Videos for Unsupervised Activity Analysis

Anonymous CVPR submission

Paper ID 210

Figure 1: In videos, each frame strongly correlates with its neighbors. Examples are sport sequences, indoor activities, surveillance scenarios or webcam streams. Our proposed approach builds a coarse to fine representation of the video sequences in a completely data-driven manner. This enables the segmentation of the video into distinct activities, as well as the interpretation of unseen sequences. We are able to detect abnormal patterns, such as the frame marked in red.

Abstract

Temporal consistency is a strong cue in continuous data streams and especially in videos. We exploit this concept and encode temporal relations between consecutive frames using discriminative slow feature analysis. Activities are automatically segmented and represented in a hierarchical coarse to fine structure. Simultaneously, they are modeled in a generative manner, in order to analyze unseen data. This analysis supports the detection of previously learned activities and of abnormal, novel patterns. Our technique is purely data-driven and feature-independent. Experiments validate the approach in several contexts, such as traffic flow analysis and the monitoring of elderly people for assisted living. The results are competitive with the state-of-the-art in all cases.

1. Introduction

The analysis of activities from videos is of utmost importance in order to solve diverse tasks such as action recognition, scene analysis or abnormal event detection (see [11, 6] for surveys). In most systems expert knowledge is required to train specific models with labeled data. Arguably, a one-time training process cannot anticipate all the possible activities, and the monitored setups may vary considerably. Hence, recent research tries to build or adapt such models automatically and in an unsupervised manner.

In previous works, human actions [14] or surveillance scenes [5, 9, 19] are analyzed automatically for the extraction of *topics* from spatio-temporal words. Their goal is to find correlated motion in order to segment behavior in space and time. Other approaches to video summarization [16, 23] cluster video streams into repeated activities. Trained models can further be used to analyze unseen behavior. In such approaches, abnormal events are often detected as outliers. This has been successfully applied to traffic monitoring [5, 7], the surveillance of public places [1], assisted living [12] or the analysis of motion patterns [15]. However, these methods often suffer from either (i) strong constraints which limit their use to specific applications, (ii) the need for prior knowledge (*e.g.*, the number of activities) and/or, (iii) being too abstract for easy interpretation.

In order to overcome these limitations, we seek for an "invariant characteristic" that can underpin generic model building and reasoning. Considering the different sequences in Fig. 1, one obvious observation is that increments between frames are quite small with respect to the changes in the whole sequence (1^{st} row). Moreover, the behavior of a tracked person (2^{nd} row) is composed of a certain repertoire of activities with transitions in between that are typically short in comparison. This can also be observed at larger scales, like day-night changes or seasonal changes (3^{rd} and 4^{th} row) and already suggests a hierarchi-

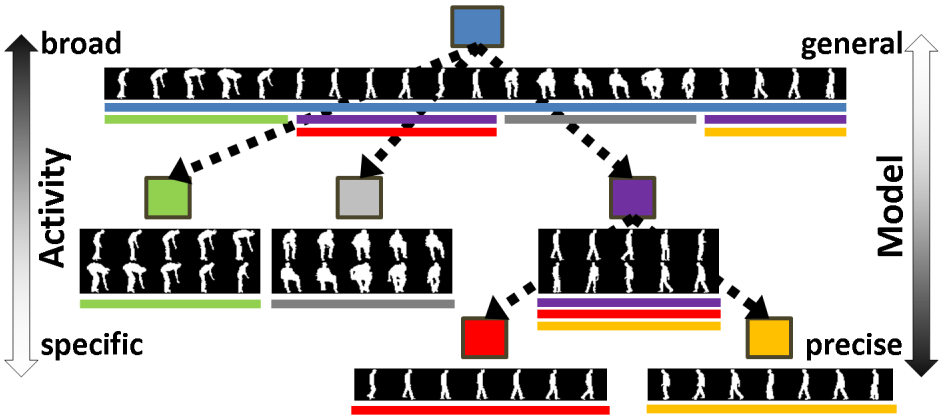


Figure 2. Overview of the proposed hierarchical model. We represent the data in a coarse to fine manner at different levels. As an example, we consider indoor actions. At the top, the behavior in general is taken into account, while at lower levels, more specific concepts, like picking up, or walking leftwards are segmented and modeled more precisely. In order to automatically split the data, temporal relations in the image stream are exploited.

cal structure.

In this work, we make use of the temporal consistency in image streams for model building. Activities are automatically discovered and continuously refined. Additionally, we are able to interpret unseen video sequences in order to redetect activities or spot abnormal events, such as the big tent in a street festival (3^{rd} row in Fig. 1).

The contributions of this paper are twofold:

- We propose an unsupervised technique to segment the data into compact and meaningful activities. To this end, we explore the strong temporal relations in the video (Sec. 2). The automatically discovered activities are efficiently represented and arranged in a hierarchical manner (Sec. 3).
- Analysis and interpretation of unseen data is demonstrated as a result of the coarse to fine representation in the hierarchy (Sec. 4).

Experimental results, presented in Sec. 5 for different video surveillance scenarios, show the usefulness and generality of the technique. We demonstrate activity segmentation, the surveillance of public places, as well as the detection of abnormalities in indoor scenarios.

2. Activities in data streams

Due to the large variety of observations in a data stream, it often is difficult to build a single model which describes the data and its dynamic behavior precisely. In this work, we automatically split the data stream into meaningful subsequences. If they are consistent and have low complexity, they can be represented more easily and precisely. This concept is similar to *motion segments* in [13] or *micro-actions* in [12]. Since we do not restrict ourselves to human actions, we call these subsequences *activities*.

From temporal relations to hierarchical models. In data obtained from time series, as in videos, there is a strong link between temporally adjacent observations. In this context, we characterize activities to have a certain duration, to be observed frequently, and to be interconnected by shorter transitions. In other words, *with high probability, neighboring frames share their activity label.* We opt to exploit this principle by arranging the video data in a hierarchical manner as outlined in Fig. 2. In a long data-stream, some activities might be very distinctive and can be segmented on a general level, while subtle concepts might need a more precise view to be detected. The advantages of our approach can be categorized as follows:

Definition of activities. Activities are automatically explored from their temporal characteristics based on discriminative modeling techniques. No prior knowledge on the boundaries or the total number or activities is required.

General vs. specific models. The dilemma between generalization capacity and precision of the model is naturally handled in the hierarchy. Nodes higher up in our hierarchical model are very general and represent a broad variety of activities (*e.g.*, "an object is moving"), whereas lower nodes only incorporate short and very specific activity patterns (*e.g.*, "a person walking to the right").

Interpretation. If the model is applied to new, unseen data at runtime, the search through the hierarchy is not only more efficient, it also allows conclusions about the nature of the unseen data. In particular, a new observation can either be assigned to a known activity or is recognized as outlier at a certain level in the hierarchy.

In the following section, we show how we establish such a hierarchical activity model from an image stream.

3. Activity summarization

Our approach is inspired by the principle of invariant or slowly varying features. Wiskott and Sejnowski [22] have proposed Slow Feature Analysis (SFA) as an unsupervised learning technique for continuous data streams, inspired by human learning capacities. Recently, Klampfl and Maass [8] have shown that SFA yields the classification capacities of Fisher's Linear Discriminant, if temporally adjacent samples in the data stream are likely to belong to the

same class. This requirement is fulfilled in our setting, as we analyze continuous streams of images and assume that activities therein are performed over a certain time span.

Given an image stream, $S = \{I_1, I_2, \ldots, I_T\}$ of T images, $I_t \in \mathbb{R}^{n \times m}$, we represent each image I_t by a D-dimensional feature vector $\boldsymbol{f}_t \in \mathbb{R}^D$. As we will show in the experiments, the feature representation is not crucial.

3.1. Data segmentation

In the segmentation step, the goal is to split the data stream into its composing activities. More generally speaking, a broader set of activities is partitioned into subsets.

Slow Feature Analysis. The output signal $\boldsymbol{z}_t$ of the Slow Feature Analysis represents the slowest components in $\boldsymbol{f}_t$, *i.e.*, it minimizes the average temporal variation:

$$\min J_{SFA} := \mathbb{E}_t(\Delta \boldsymbol{z}_t), \text{ where } \Delta \boldsymbol{z}_t = ||\boldsymbol{z}_t - \boldsymbol{z}_{t-1}||^2. \quad (1)$$

To avoid the trivial solution $\boldsymbol{z} \equiv 0$, additional constraints for zero mean and unit variance are introduced. Multiple slow features need to be decorrelated and they are ordered by decreasing *slowness*.

Let $\boldsymbol{y}_t = \boldsymbol{f}_t - \mathbb{E}_t(\boldsymbol{f}_t)$ be the zero-mean feature vector. Considering only linear functions of the form $\boldsymbol{z} = \boldsymbol{w_s}^\mathsf{T} \boldsymbol{y}$, it can be shown [21] that the objective becomes

$$\min J_{SFA}(\boldsymbol{w}_s) := \frac{\boldsymbol{w}_s^\mathsf{T} \dot{\boldsymbol{D}} \boldsymbol{w}_s}{\boldsymbol{w}_s^\mathsf{T} \boldsymbol{D} \boldsymbol{w}_s}, \quad (2)$$

where $\boldsymbol{D} = \mathbb{E}_t(\boldsymbol{y_t} \boldsymbol{y_t}^\mathsf{T})$ is the covariance matrix of the data and $\dot{\boldsymbol{D}} = \mathbb{E}_t\big((\boldsymbol{y_t} - \boldsymbol{y}_{t-1})(\boldsymbol{y_t} - \boldsymbol{y}_{t-1})^\mathsf{T}\big)$ the covariance matrix of the temporal differences.

The weight vectors $\boldsymbol{w}_s$ which minimize Eq. (2) are the solutions to the generalized eigenvalue problem

$$\dot{\mathbf{D}} \boldsymbol{w}_s = \lambda_s \boldsymbol{D} \boldsymbol{w}_s. \quad (3)$$

The slowest varying components in $\boldsymbol{y}$ are their projections onto the eigenvectors $\boldsymbol{w}_s$ associated to the smallest eigenvalues λ_s [21].

Clustering. In the SFA subspace, distinct activities are discriminatively clustered into distinct high density regions with sparse transitions [8]. Hence, we apply Gaussian Mixture Model (GMM) clustering to separate the activities. By means of expectation maximization, the regions where the data is densly scattered are found. The cluster index assigned to a data point corresponds to the cluster number with maximal posterior probability [2]. Initialization is done with *k-means*. Since the desired number of clusters is not known a priori, a sweep over k is performed and the sum of posterior probabilities over all datapoints is calculated. The second derivative of this sum characterizes the curvature and we select its maximum as the desired number of clusters. A postprocessing step ensures temporal smoothness and discards very short sequences.

3.2. Building the activity hierarchy

The segmentation is applied recursively on the data. In the first step, we split according to the most dominant (slowest) cues in the entire datastream. In order to create a hierarchy, the segmentation process is repeated for each obtained subset and other discriminative components may now appear. This is encouraged since we keep the dimensionality of the SFA subspace fixed. Repeating the segmentation sets up a hierarchical structure. At high levels, the established nodes contain very broad activity concepts while at lower levels in the hierarchy, specific actions are segmented.

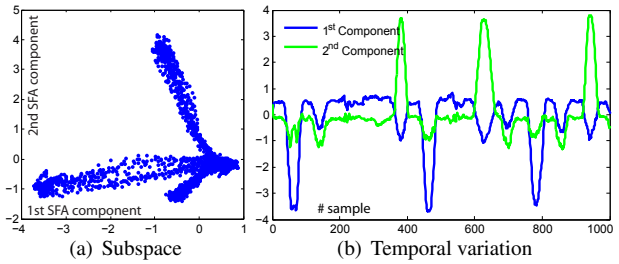
Basic activities. The decision whether or not a node is further refined is based on the representation in the SFA space. The data is projected so that the average distance between consecutive samples is minimized, *c.f.* Eq. (1). If the distances are approximately equal across the whole sequence, the data is well described by its slowest components [22]. In this case, we define a *basic activity* A and the data is not split any further. This corresponds to a leaf node in the hierarchy. On the other hand, if major parts of the data are connected with short distances in the subspace, there must be a few consecutive samples which lie far apart, due to the unit variance constraint. This setting verifies the discriminative assumption of [8], hence, splitting the data is stimulated.

As a simple measure of data compactness, we use the median of distances between consecutive samples in the SFA space. It turns out to be robust against outliers, and reflects well the concept above. If we measure a small median value, the data is further segmented. For a larger median, a basic activity A is detected.

Illustration. To get an intuition of our summarization technique, we now discuss it with respect to the activity dataset from [16] and show how our results compare to theirs. We use silhouette data from two views as provided by the authors, apply a distance transform on both and concatenate the rows to one feature vector. The data exhibits five actions (*throw, bend, squat, bat, pick phone*). Each of them is repeated ten times with different execution speeds. We randomly permutate the actions and the repetitions and consider them as one input video to be segmented.

The data is analyzed by means of SFA and the first two dimensions of the resulting manifold are displayed in Fig. 3(a). In Fig. 3(b), the variations of these two slowest components are plotted over time. The subspace clearly holds discriminative characteristics.

In Fig. 4, the first two dimensions of the clustered three-dimensional SFA subspace are displayed. It is obtained at the root node, where the data of all five actions is included. As seen in the sketched hierarchy, four basic activities are already extracted after the first split. The pink node is subdivided. In Fig. 5 the segmentation criterion is verified. The



(a) Subspace (b) Temporal variation

Figure 3. The data projections (a) and the evolution of the slowest components (b) characterize the discriminative SFA subspace.

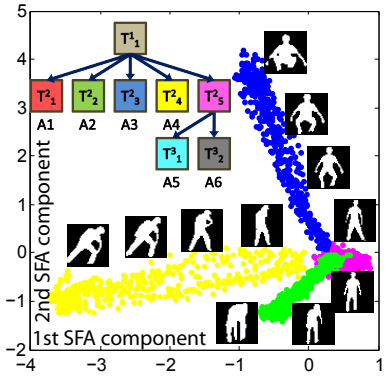


Figure 4. GMM clustering in the SFA subspace, viewed in two dimensions. The resulting activity hierarchy is sketched.

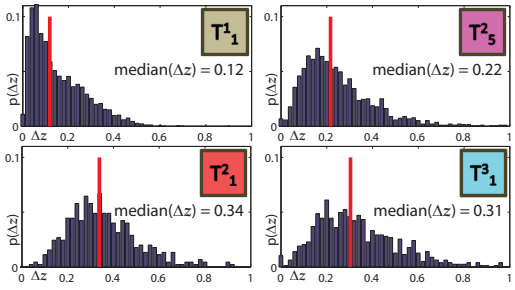


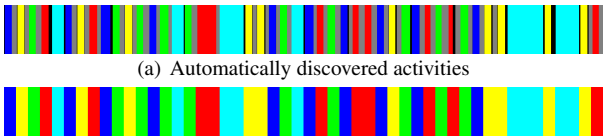
Figure 5. Segmentation criterion based on the distribution of distances between consecutive samples in the SFA space. For basic activities, the median is higher, and they are not further segmented.

empirical distribution of distances $\Delta \boldsymbol{z}_t$ is shown and their medians are indicated. For nodes T_1^1 and T_5^2, there is a clear shift of the mode towards the origin, hence they are split further. This is not the case for leaf nodes.

From the segmentation process, six basic activities are automatically discovered, they precisely correspond to the five ground truth actions ($A1 - A5$) plus the standing still activity ($A6$). The latter is at the beginning and at the end of each performed action. The samples that were filtered in the modelling procedure (short sequences and outliers) are collected in $A0$. The summarization results are reported in Tab. 1 and Fig. 6. If we neglect the standing activity $A6$ as well as $A0$, we notice a perfect match between our result and the ground truth, and we outperform [16]. This demonstrates the great benefit of segmentation in the discriminative SFA subspace.

Activity	$A0$	$A1$	$A2$	$A3$	$A4$	$A5$	$A6$
Throw	2(.05)	10(.58)	0	0	0	0	10(.37)
Bend	3(.03)	0	10(.60)	0	0	0	10(.37)
Squat	8(.07)	0	0	10(.61)	0	0	10(.32)
Bat	10(.17)	0	0	0	10(.57)	0	9(.26)
Phone	3(.01)	0	0	0	0	10(.99)	1(.00)

Table 1. Composition of the automatically discovered basic activities ($A0 - A6$) *vs.* the ground truth. The number of occurrences are counted, and the percentage of number of frames is reported in brackets. $A1 - A5$ perfectly match the ground truth.



(a) Automatically discovered activities

(b) Ground truth activities

Figure 6. Color coded labeling for the discovered and the ground truth actions. The activity data from [16] is used.

3.3. Data modeling

As we want to use the hierarchy to classify the activities in previously unseen videos, the data underlying each of its nodes is modeled with respect to shape and dynamics. Biological studies on human motion perception suggest that motion analysis is performed from a sequence of time varying appearance snapshots [10]. We therefore create an extended feature vector $\boldsymbol{v}_t = (\boldsymbol{f}_t, \boldsymbol{f}_{t-1}, \dots \boldsymbol{f}_{t-n})^\mathsf{T}$ as the concatenation of the last n feature representations, similar to [17]. Let further $\boldsymbol{x}_t = \boldsymbol{v}_t - \mathbb{E}_t(\boldsymbol{v}_t)$ be the zero-mean feature vectors to be encoded.

Principal Component Analysis. PCA is a well known technique for low dimensional data representation. In order to maximally capture the information in the D-dimensional data, PCA projects this data into a linear subspace which maximizes the variance [2], *i.e.*

$$\max J_{PCA}(\boldsymbol{w}_p) := \mathrm{Var}_t(\boldsymbol{w}_p^\mathsf{T}\boldsymbol{x}_t) = \boldsymbol{w}_p^\mathsf{T}\boldsymbol{C}\boldsymbol{w}_p \qquad (4)$$

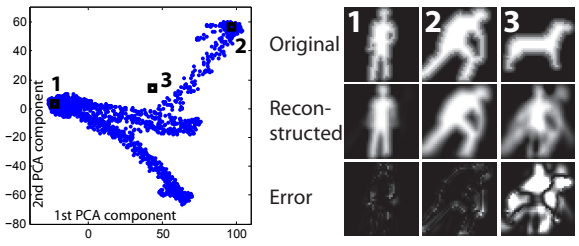
where $\boldsymbol{C} = \mathbb{E}_t(\boldsymbol{x}_t\boldsymbol{x}_t^\mathsf{T})$ is the data covariance matrix. Again, additional constraints on unit variance and orthonormality exclude trivial solutions. PCA can also be formulated in terms of minimizing the mean reconstruction error:

$$e = \mathbb{E}_t\big((\boldsymbol{x}_t - \boldsymbol{x}_t^*)^2\big), \quad \text{where} \quad \boldsymbol{x}_t^* = \sum_{i=1}^{d} a_{t,i}\boldsymbol{w}_{p,i}. \qquad (5)$$

Keeping only the first $d < D$ principal components compresses the data. The reconstructed datapoint $\boldsymbol{x}_t^*$ is then an approximation of the original $\boldsymbol{x}_t$, but relies only on the d-dimensional representation a_t. The general solution to Eq. (4) is obtained by solving the eigenvalue problem

$$\boldsymbol{w}_p = \lambda_p \boldsymbol{C}\boldsymbol{w}_p. \qquad (6)$$

The eigenvectors $\boldsymbol{w}_p$ that correspond to the d largest eigenvalues λ_p are selected as projection basis. In Fig. 7(a), the



(a) Subspace (b) Illustration

Figure 7. (a) Two principal components in the PCA subspace. (b) Two training images and one outlier projected and reconstructed.

two dimensional PCA manifold of the activity data at the root node T_1^1 is shown. As seen from Fig. 7(b), this model represents well the training data, but has a high reconstruction error for an unfamiliar shape.

The formulations of SFA and PCA are similar, however, PCA creates a generative data model while SFA encodes differences and delivers a discriminative characterization.

Tracker hierarchy. Since form and motion are modeled jointly, every node *follows* the data and is in fact a tracker T_i^j (the i-th tracker on level j). To enforce the idea of general *vs.* specific trackers, we keep the dimensionality d of the PCA subspace fixed for each of them. If a node high up in the hierarchy describes its large amount of data with the same number of dimensions d as a node specifically tuned to one activity, the latter incorporates a much preciser representation. Hence, we dispose of a set of more general and more specific trackers.

If a basic activity is detected, we add one additional PCA model to this leaf node. We abandon the fixed dimensionality d and increase it to $d' > d$ to capture a certain percentage of the total variance in the subspace. This tunes this node even more specifically to the segmented basic activity A.

4. Analysis of unseen data

Once the training data has been modeled as just described, the hierarchical model can be applied for the interpretation of new data. In this section we show how to detect and classify activities as well as how to report abnormal activities at multiple levels.

4.1. Activity classification and tracking

Given a new sequence $\boldsymbol{x}'$ and a set of basic activities $\mathcal{A}$, the task is to identify $A \in \mathcal{A}$, which best explains the data. To this end, $\boldsymbol{x}'$ is projected into the corresponding PCA subspaces and the reconstruction errors are calculated as in Eq. (5). The tracker T_A with be lowest reconstruction error e_{T_A} defines the discovered activity

$$A^\star = \arg\min_A e_{T_A}(\boldsymbol{x}'). \tag{7}$$

The hierarchical description makes sure that not all PCA models need to be tested, as discussed in the next section.

Simultaneous tracking and activity reasoning In certain applications, only a sub-region of the entire scene might be considered, for example if the activities of a person are analyzed. Tracking the person throughout the video sequence means determining his location within each image. This search for an optimally estimated location can be incorporated in the previous formulation. Each activity is evaluated at various image locations and scales ρ, and the according reconstruction error $e_{T_A}(\boldsymbol{x}'|\rho)$ is obtained. In an exhaustive search procedure the optimal location and activity are determined simultaneously:

$$(\rho^\star, A^\star) = \arg\min_{A,\rho} e_{T_A}(\boldsymbol{x}'|\rho). \tag{8}$$

For efficiency reasons and since temporal consistency is assumed in tracking, only the local neighborhood of ρ^*_{t-1} is scanned.

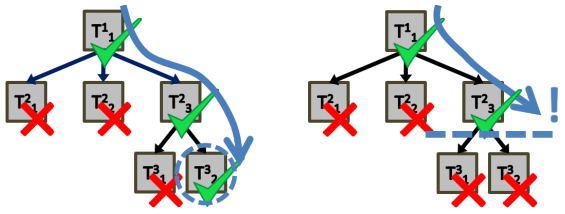
4.2. Hierarchical interpretation

We now show how the hierarchical model paves the way for a more sophisticated and efficient analysis. Since the hierarchy consists of a set of more general and more specific trackers, we can apply the anomaly reasoning as proposed in [20]. In their terminology, we dispose of a disjunctive hierarchy on multiple layers and every more general node has a number of more specific sub-nodes. For a consistent reasoning, we need to transform each tracker into a classifier. Any tracker T_i^j in the hierarchy is considered *active* for an observation $\boldsymbol{x}'$ based on its normalized reconstruction error:

$$\text{active}(T_i^j) = \begin{cases} 1 & \text{if } \dfrac{e_{T_i^j}(\boldsymbol{x}') - \mu_{T_i^j}}{\sigma_{T_i^j}} < \theta \\ 0 & \text{otherwise} \end{cases}, \tag{9}$$

where $\mu_{T_i^j}$ and $\sigma_{T_i^j}$ are respectively the mean and the standard deviation of the reconstruction error for tracker T_i^j obtained during training.

As long as the observations are according to expectations, there is always a leaf node tracker (*i.e.* basic activity) which is able to explain the data. To respect the hierarchy, each observation is propagated from the root node to the leaves as sketched in Fig. 8(a). Only subnodes of active



(a) Valid activity (b) Abnormal event at level 3

Figure 8. Interpretation of new data within the learned hierarchical model. (a) Known activities, represented by leaf node trackers are detected. (b) A reasoning on abnormal conditions in the hierarchy is performed from active and inactive trackers on different levels.

trackers need to be considered, which additionally increases the efficiency.

If a more general node tracker validates the observations, but none of its more specific sub-node trackers does, then this signals an abnormal activity (Fig. 8(b)). Such abnormality can occur at any level. From the location in the hierarchy where this happens, interpretations about the nature of the abnormality can be made.

5. Experiments

5.1. Implementation details

Noise reduction. As we deal with noisy data in practice, noise reduction is applied in the training phase. At every node in the hierarchy we apply PCA and keep the amount of dimensions in order to describe 95% of the the total data variance. Segmentation (SFA) and modelling (PCA) is performed thereafter. Furthermore, at each step down the hierarchy, we eliminate samples that were outliers in the PCA space (*i.e.* $e > \mu_{T_i^j} + 3\sigma_{T_i^j}$). This is necessary, as the training data is not assumed to be free from anomalies.

Parameters. We use the following parameters: SFA and PCA subspaces are modeled in $d = 3$ dimensions and $n = 5$ last frames are used for motion encoding. PCA models of basic activities keep 80% of the data variance, which results in most cases in $d' = 5$ to 10 dimensional subspaces. For hierarchical reasoning, the threshold $\theta = 3$ is applied. We also tested different parameter sets, without observing significant changes.

Runtime. Due to its low complexity, the runtime analysis is performed in real-time for surveillance scenarios. On a standard PC, our current matlab implementation runs at more than 10 frames per second. The exhaustive search in the person tracking case slows down evaluation by approximately a factor of 10. The time consumption model building is in the order of a few minutes for our cases.

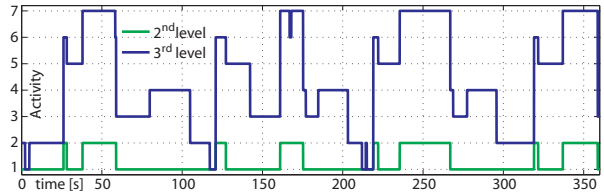
5.2. Surveillance of public places

We show how our technique performs on three different visual surveillance datasets.

QMU junction [5] (360×288 pixels, 25 fps, 1 hour). This data has previously been used for learning spatio-temporal scene topics [5]. We show how our approach can produce similar results but by using a holistic scene descriptor. As car and pedestrian motion is of importance in this scene, we use motion monitors [18] augmented with a motion history factor. We use 18×18 pixel patches and a forgetting rate of 0.95. Training is done on $50,000$ images, the runtime evaluation takes into account all $90,000$ frames. The model extracts 19 basic activities within the 48 nodes.

Some discovered basic activities are depicted in Fig. 9(a). In fact, the hierarchical model nicely groups consistent traffic patterns in leaf nodes. Other activities characterize empty streets with pedestrian motion, cars accelerating, and different turn configurations. In Fig. 9(b) we show the obtained activity segmentation over time, for the second and third level in the hierarchy. Without enforcing any larger scale temporal relations, we discover pseudo-repeated patterns in the data that correspond to different traffic activities. As successfully done for example in [9], these patterns can be additionally learned for the detection of irregular ordering of different familiar activities.

(a) Discovered activities: Driving left to right, right to left, straight



(b) Activities over time at the first two levels in the hierarchy

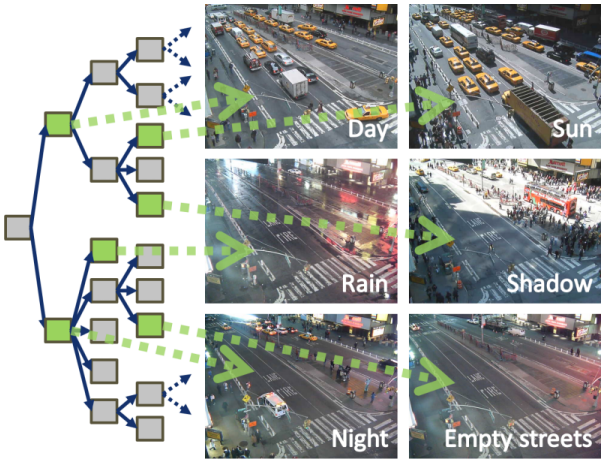
(c) Anomalies: Ambulance, bus and truck collision course, wrong direction)

Figure 9. Activities, their temporal variation and detected anomalies for the QMU junction [5]. Regions with high reconstruction error are shaded in red. (Figures are best viewed in color.)

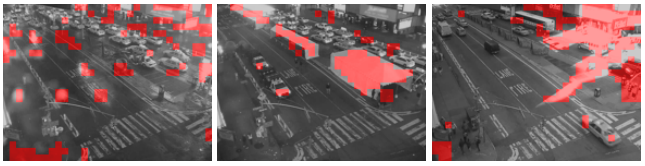
Three exemplary abnormal situations are presented in Fig. 9(c), two of which have also been reported in [5]. Since our technique analyzes the entire scene at once, unseen configurations, like the middle example, are also reported as abnormal. Among all the detected abnormal events, there are hardly any that have no plausible interpretation.

HUJI street crossing [4] (320×240 pixels, 10 fps, 2 hours). Without modifications from above, we apply our technique to the street crossing data from [4]. The model is trained on the first hour and has 27 nodes of which 11 are basic activities. A selection of abnormal events from the second hour is reported in Fig. 10.

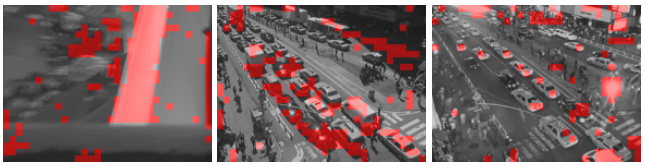
Figure 10. Anomalies on the HUJI crossing [4]: Taxi driving backwards, collision course, driving on pedestrian crossing.



(a) Discovered activities and semantic interpretation in the hierarchy



(b) Anomalies (same as [3]): Heavy rain, festival tent, shadow shape.



(c) Anomalies (other): Camera failure, jam with reflections, strong light.

Figure 11. Activities and anomalies detected for the Times Square webcam data [3].

Times Square [3] (640×480 pixels, approximately 0.3 fps, 2 months). In this dataset, images from a webcam overlooking Times Square in New York are taken at low frame rate over a long period. The relation between adjacent frames is thus to be handled on a larger time scale. We simply describe the original color images in downsampled (24×32 pixel) grayscale images. Due to the low frame rate, no motion is included in the PCA model.

We construct the hierarchical model with data from 17 days ($150,000$ images, every 3rd image). The obtained hierarchy has 65 tracker nodes, thereof 26 basic activities. In Fig. 11(a), we display the tree-like structure for the first four levels, and show typical instances of second level and leaf nodes. Apparently, day-night changes are the most dominant cues, which are segmented in the first step. The obtained activities are mostly well interpretable, as the provided examples show.

In Fig. 11(b) and (c) we show six illustrative abnormal events that are detected among more than $250,000$ evaluated frames. In general, we detect similar anomalies as reported in [3], as the ones in Fig. 11(b). In addition to their work, we also report many cases of incomplete frames, camera failures, water on the lens and other salient situations.

5.3. Human behavior analysis

In this section a person is being tracked within the scene and simultaneously his behavior is analyzed. We use the data from [12] (640×480 pixels, 15 fps) and their silhouette features. The training video is provided with normal daily activities, and the model is evaluated on a test sequence that contains abnormal events, motivated for monitoring elderly people in their homes.

We trained the model from $7,100$ frames in the training set. The obtained hierarchy is visualized in Fig. 12, and for each leaf node tracker, the corresponding activity is shown. The model nicely encodes the different aspects of human behavior. At higher levels, it distinguishes between upright and other poses, at low levels, sitting, picking up, walking leftwards or rightwards are segmented as basic activities.

Results and comparison. The hierarchy is applied to the test video which consists of $1,030$ frames. In Fig. 13(a-e) some selected frames from this sequence are displayed, they show three normal activities and two detected anomalies. The observed person is tracked in space and the matching activity is determined simultaneously. The plot on the lower part of Fig. 13 characterizes the evolution of the tracker membership over time. The y-axis in this plot indicates the number that training has assigned to the ongoing basic activity (Fig. 12). $A0$ groups the outliers. At each frame, the activity best explaining the observations, is reliably determined by our tracking method.

We quantitatively compare the overall performance of our approach to the results of [12]. The recall - precision curve is obtained by sweeping the parameter θ which determines whether a tracker is active or not. The obtained characteristic is displayed in Fig. 14. Our technique clearly outperforms the previous state-of-the-art, in particular we increase the recall from 68% to approximately 83% at 99% precision.

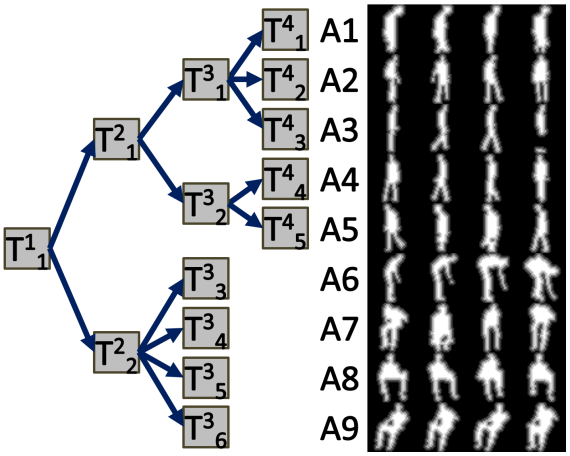


Figure 12. Learned hierarchical model: The nine emerged basic activities are visualized in correspondence to the leaf nodes.

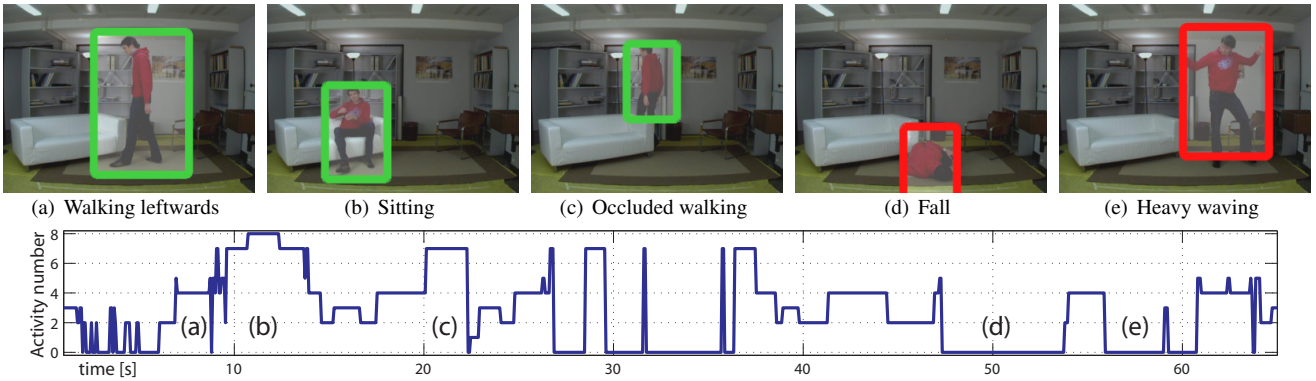


Figure 13. (a-e) Sample frames which illustrate detected familiar activities and abnormal events. In the lower part, the detected activities are reported over time. The numbering of basic activities on the y-axis is the same as in Fig. 12, $A0$ corresponds to an anomaly. The location of frames (a-e) is indicated. **Please refer to the supplemental material for videos.**

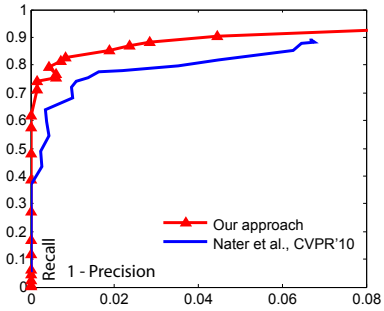


Figure 14. Our approach significantly outperforms previous state-of-the-art in terms of RPC in the in-house scenario.

6. Conclusion

In this paper, we presented a data-driven approach to activity segmentation that exploits temporal relations present in video sequences. The small changes from frame to frame are discriminatively analyzed, in order to set up a powerful hierarchical model. We have shown how this model is applied to unseen videos at runtime and that the hierarchy can be used to explain the observations. Due to two linear techniques of low computational complexity (SFA for clustering and PCA for modeling) we are able to efficently detect normal and abnormal activities. Finally, qualitative and quantitative results demonstrate the validity of our technique.

References

[1] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz. Robust real-time unusual event detection using multiple fixed-location monitors. *PAMI*, 30(3):555–560, 2008.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.

[3] M. D. Breitenstein, H. Grabner, and L. Van Gool. Hunting nessie – real-time abnormality detection from webcams. In *ICCV WS on Visual Surveillance*, 2009.

[4] A. Hendel, D. Weinshall, and S. Peleg. Identifying surprising events in video using bayesian topic models. In *Proc. ACCV*, 2010.

[5] T. Hospedales, S. Gong, and T. Xiang. A markov clustering topic model for mining behaviour in video. In *Proc. ICCV*, 2009.

[6] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34(3):334 –352, 2004.

[7] W. Hu, X. Xiao, Z. Fu, D. Xie, F.-T. Tan, and S. Maybank. A system for learning statistical motion patterns. *PAMI*, 28(9):1450–1464, 2006.

[8] S. Klampfl and W. Maass. Replacing supervised classification learning by Slow Feature Analysis in spiking neural networks. In *NIPS*, 2009.

[9] D. Kuettel, M. D. Breitenstein, L. Van Gool, and V. Ferrari. Whats going on? Discovering spatio-temporal dependencies in dynamic scenes. In *Proc. CVPR*, 2010.

[10] J. Lange and M. Lappe. A model of biological motion perception from configural form cues. *Journal of Neurosciences*, 26:2894–2906, 2006.

[11] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *CVIU*, 104(2):90–126, 2006.

[12] F. Nater, H. Grabner, and L. Van Gool. Exploiting simple hierarchies for unsupervised human behavior analysis. In *Proc. CVPR*, 2010.

[13] J. C. Niebles, C.-W. Chen, , and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *Proc. ECCV*, 2010.

[14] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *IJCV*, 79:299–318, 2008.

[15] C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *PAMI*, 22(8):747–757, 2000.

[16] P. Turaga, A. Veeraraghavan, and R. Chellappa. Unsupervised view and rate invariant clustering of video sequences. *CVIU*, 113(3):353–371, 2009.

[17] R. Urtasun, D. J. Fleet, and P. Fua. Temporal motion models for monocular and multiview 3d human body tracking. *CVIU*, 104(2):157–177, 2006.

[18] G. Veres, H. Grabner, L. Middleton, and L. Van Gool. Automatic workflow monitoring in industrial environments. In *Proc. ACCV*, 2010.

[19] X. Wang, X. Ma, and W. Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *PAMI*, 31(3):539 –555, 2009.

[20] D. Weinshall, H. Hermansky, A. Zweig, J. Luo, H. Jimison, F. Ohl, and M. Pavel. Beyond novelty detection: Incongruent events, when general and specific classifiers disagree. In *NIPS*, 2008.

[21] L. Wiskott. Slow feature analysis: A theoretical analysis of optimal free responses. *Neural Computation*, 15(9):2147–2177, 2003.

[22] L. Wiskott and T. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, 14(4):715–770, 2002.

[23] F. Zhou, F. De la Torre, and J. F. Cohn. Unsupervised discovery of facial events. In *Proc. CVPR*, 2010.