Exploiting Simple Hierarchies for Unsupervised Human Behavior Analysis

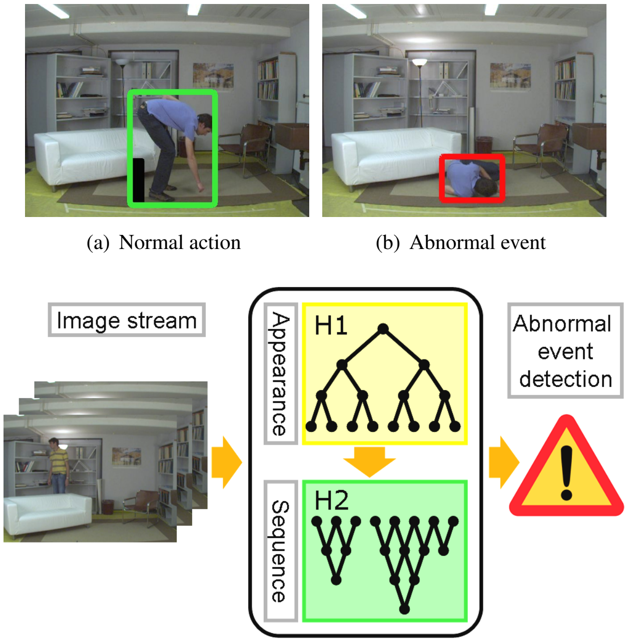
Fabian Nater[1] Helmut Grabner[1] Luc Van Gool[1,2]

[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.*



(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.

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 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. 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.

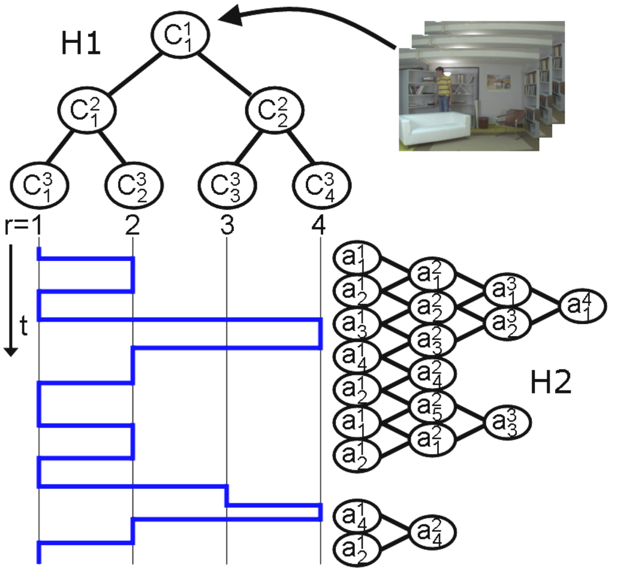


Figure 2. Illustration of the unsupervised learning approach, composed of two hierarchies. In $H1$, a sequence of images is mapped by clustering to a number of discrete symbols, in $H2$ the sequence of these symbols is analyzed.

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.

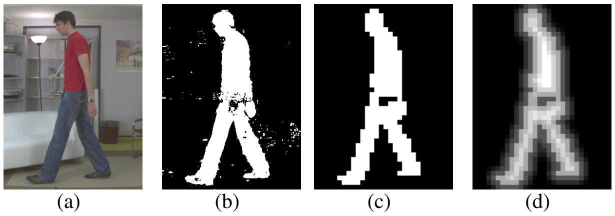

(a) (b) (c) (d)

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 $x_v(n)$, $n = 1 \ldots N$ are at a distance

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

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.

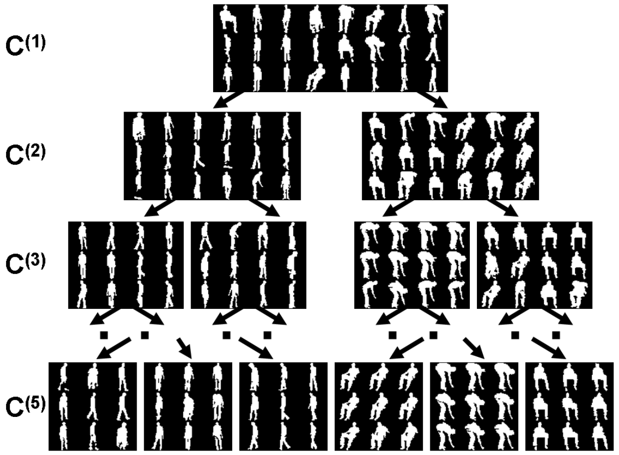


Figure 4. Visualization of the proposed binary tree for the hierarchical appearance representation ($H1$). For each of the displayed clusters at different layers $C_i^{(l)}$, randomly chosen silhouettes are displayed.

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:

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

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.*

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

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 $\langle r_1, \ldots, r_t \rangle$ by frequent patterns a_i^{λ} and we can represent the image stream as a series of *micro-actions* of different lengths λ:

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

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

[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.

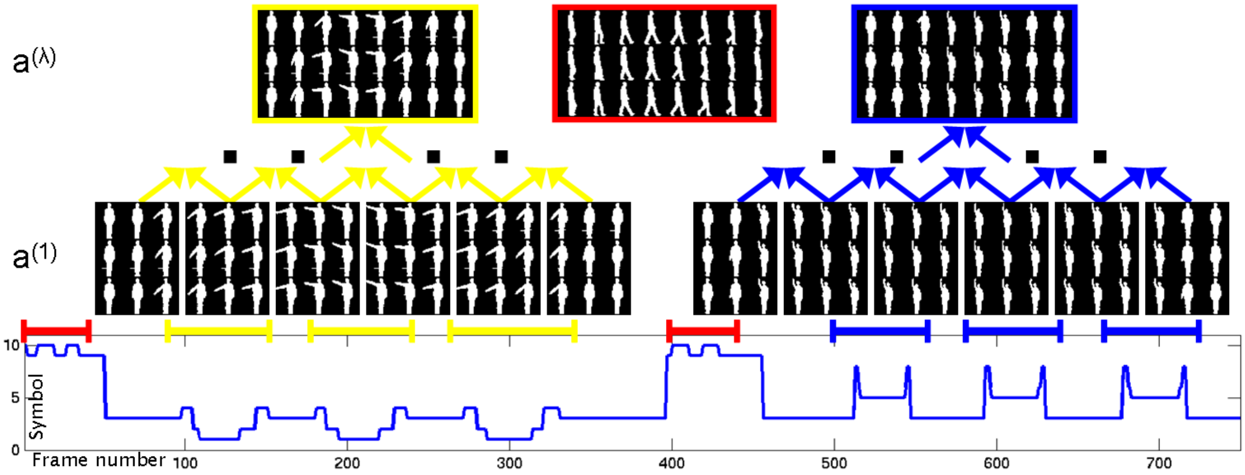


Figure 5. Illustration of the micro-action hierarchy ($H2$) for the action recognition test dataset [13]: Micro-actions are extracted from symbol transitions and can be combined gradually into higher level actions.

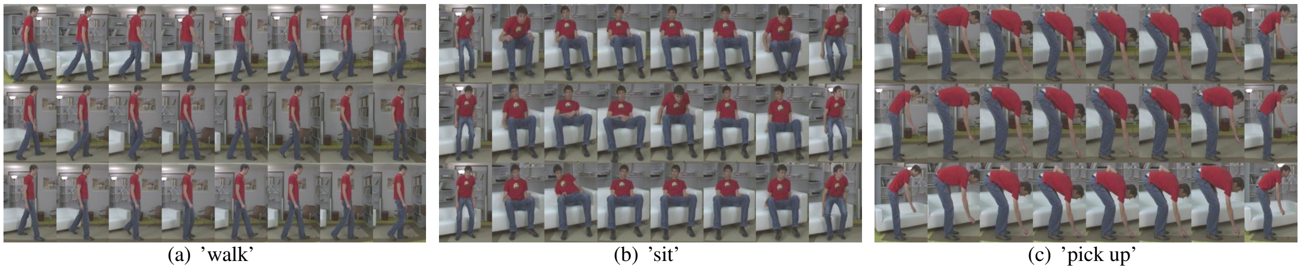


Figure 6. Examples of segmented actions as produced with our method. In an unsupervised manner repetitive microactions are extracted, which can be labeled manually, if desired. Repetitions in the training dataset are presented in rows.

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 **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 **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

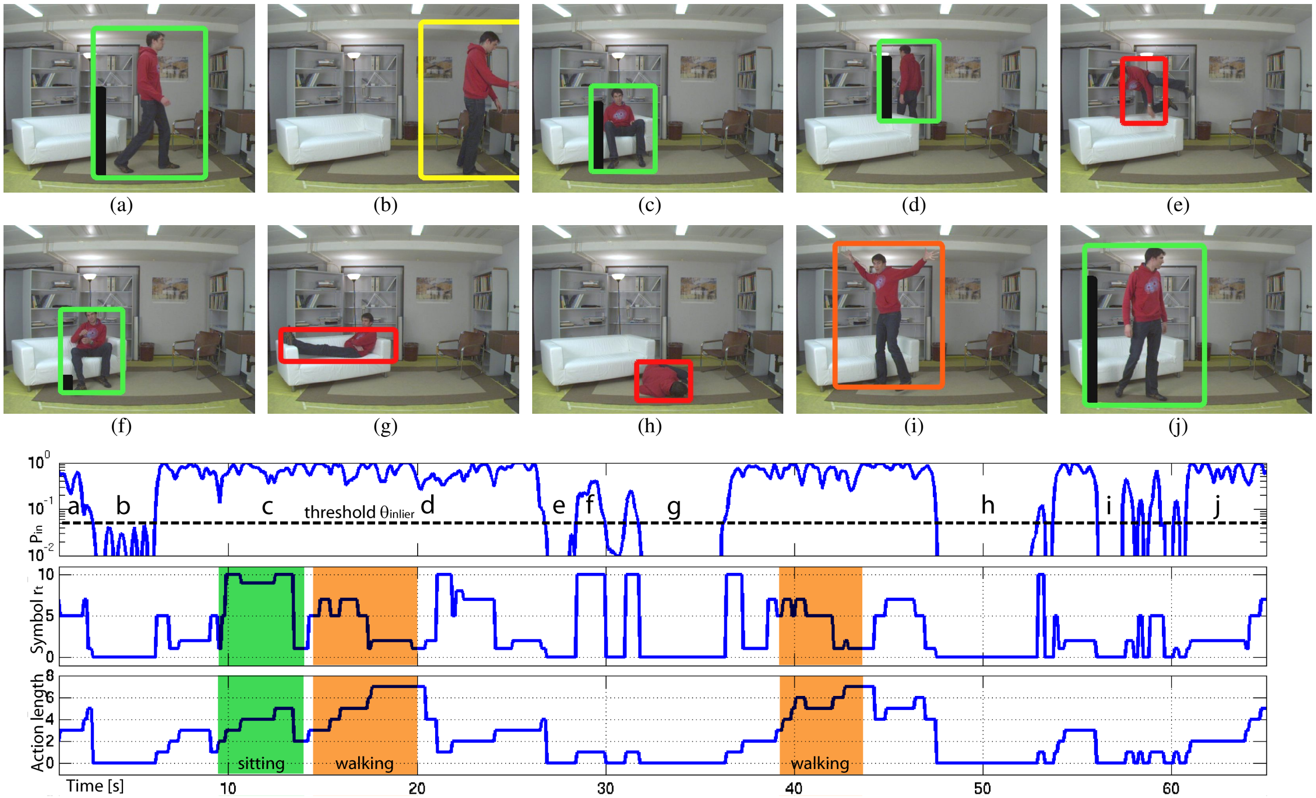
The test footage of about $1,000$ images was recorded in the same setting as the training sequence ($c.f.$ Sec 2), but now contains abnormalities such as heavy waving, jumping over the sofa and a fall. The model of normality was established as explained previously (appearance clustering in Fig. 4, extraction 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


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

(f) (g) (h) (i) (j)

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)).

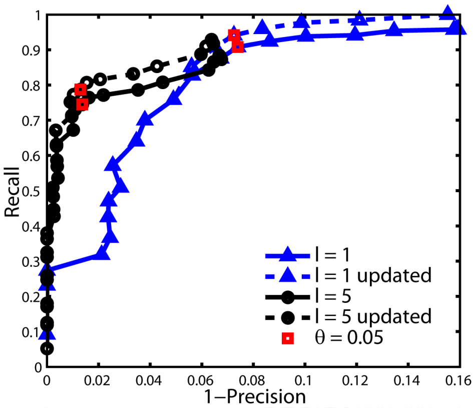
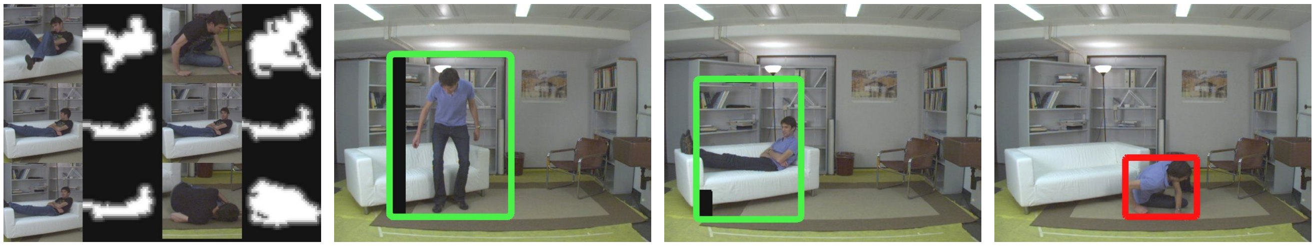


Figure 9. Recall-precision curves for the video sequence of Fig. 7 verifies the applicability of our technique.

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



(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.