DISS. ETH NO. 20377

Abnormal Behavior Detection in Surveillance Videos

A dissertation submitted to

ETH ZURICH

for the degree of

Doctor of Sciences

presented by

Fabian Nater
Ing. él. dipl. EPF
born June 14, 1981
citizen of Kemmental (TG), Switzerland

Examination Committee:

Prof. Dr. Luc Van Gool, ETH Zürich and K.U. Leuven, examiner
Prof. Dr. Daphna Weinshall, Hebrew University of Jerusalem, co-examiner
Dr. Barbara Caputo, Idiap Research Institute, co-examiner

2012

Abstract

In recent years, the number of surveillance cameras installed to monitor private and public spaces and areas has increased dramatically. To a large extent, the currently applied tools for an automated analysis detect precisely pre-defined concepts of abnormal behavior, such as an intruder in a prohibited zone. Alternatively, the video streams are constantly recorded or observed by operators. For many visual surveillance applications however, an ideal analysis software would automatically interpret the entire scene and alert in case of any suspicious situation.

The developments presented in this thesis aim at an automatic and independent detection of abnormal behavior in surveillance videos. To this end, we follow the indirect route of modeling the frequently observed normal behavior, and detect abnormal events as outliers to these models of normality. We show that encoding the video data in hierarchical models is extremely useful with respect to anomaly detection performance and semantic interpretability of the spotted abnormal events.

We present four different methods. In a first part, we propose tracker-trees as a supervised manner to arrange more and less specialized trackers in a tree-like structure and show how to interpret their relative outputs. Due to recent transfer-learning techniques, we identify in a second step how the training effort for the tracker-tree can significantly be reduced. In order to adapt more precisely to the characteristics of a particular scenario, self-learning techniques are called for. Accordingly, we develop an unsupervised approach that is inspired by recent biological findings and models the observed human behavior in distinct hierarchies for appearance and motion. One step further, we finally demonstrate that the incorporation of the temporal characteristics in activities leads to preciser and more interpretable behavior models which are still learned in a data-driven manner.

All the approaches to model normal behavior are clearly geared towards the detection of abnormal situations. In particular, we are interested in autonomous living scenarios, where (elderly) people are living on their own and an automated alert system would greatly improve their personal safety. Thanks to the generic characteristics, our techniques are widely applicable and we additionally show the use for webcam or traffic analysis and industrial workflow monitoring.

Zusammenfassung

Videoüberwachungssysteme werden seit einigen Jahren systematisch in privaten und öffentlichen Räumen und Plätzen eingesetzt. Die Anwendungsszenarien sind vielseitig. In vielen Fällen soll die Sicherheit von Personen verbessert werden, zum Teil auch präventiv. Kameras werden zum Beispiel auch verwendet, um Verkehrskontrollen durchzuführen, Menschenströme zu analysieren oder Eindringlinge zu erkennen. In der Praxis werden die Videodaten oft von ausgebildetem Personal konstant überwacht, oder zur nachträglichen Kontrolle auf Speichermedien aufgezeichnet.

Eine automatische Warnung bei abnormalen oder unerwarteten Ereignissen ist für vielen weiterreichende Anwendungen jedoch wünschenswert oder sogar notwendig. In der vorliegenden Doktorarbeit erarbeiten wir Methoden, die unabhängig von menschlichem Zutun eine Szene analysieren, Modelle von normalen Aktivitäten in dieser Szene erstellen und so ausserordentliche Ereignisse erkennen können. Wir zeigen, dass diese indirekte Beschreibung des Abnormalen Vorteile mit sich bringt in Bezug auf eine flexible Anwendbarkeit sowie eine Erkennung von verschiedenen Ereignissen, die nicht von vornherein spezifiziert werden müssen. In allen Methoden setzen wir hierarchische Modelle ein, die eine gewisse semantische Interpretation der abnormalen Ereignissen ermöglichen.

Wir stellen vier verschiedene Ansätze vor. Erstens werden in den Tracker-Trees mehr oder weniger spezialisierte und manuell trainierte Trackers hierarchisch angeordnet. Diese werden parallel ausgewertet, und auf Grund von gegenseitigen Widersprüchen können abnormale Situationen erkannt werden. In einer zweiten Phase zeigen wir wie die einzelnen Trackers mit minimalem Aufwand trainiert werden. Da jedoch viele Überwachungsszenen einzigartig sind, ist es unerlässlich, dass sich die Modelle genau der Szene anpassen. Die dritte Methode ist darauf ausgerichtet, Modelle von normalem Verhalten ohne menschliche Hilfe zu erstellen. In einem biologisch motivierten, zweistufigen Verfahren

werden die Bewegungen des beobachteten Menschen automatisch modelliert. Schlussendlich zeigen wir wie der explizite Einbezug von zeitlichen Zusammenhängen in ein solches Modell eine automatische Erkennung von interpretierbaren Tätigkeiten ermöglicht, und zugleich die Erkennungsrate von abnormalen Ereignissen verbessert.

Die meisten Experimente in dieser Doktorarbeit sind ausgerichtet auf eine automatische Erkennung von abnormalen und gefährlichen Situationen in Wohnräumen von (älteren) Menschen. Stürze zu detektieren ist ein Ziel, aber auch andere Ungereimtheiten sollen aufgedeckt werden, wie zum Beispiel wenn die Person plötzlich anfängt zu hinken. Damit könnte die persönliche Sicherheit dieser Menschen erheblich gesteigert werden. Da die vorgestellten Techniken aber nicht anwendungsspezifisch sind, zeigen wir auch, wie sie zur Verkehrsüberwachung, zur Analyse von Webcam-Daten und zur Interpretation von Arbeitsabläufen in der Industrie eingesetzt werden können.

Acknowledgements

This thesis would not have been possible without the great support, inspiration and influence of many persons. My greatest thanks go to:

Helmut Grabner ⋆ Luc Van Gool.

Daphna Weinshall ⋆ Barbara Caputo.

Tobias Jaeggli ⋆ Severin Stalder ⋆ Henning Hamer ⋆ Tobias Gass ⋆ Philipp Fuernstahl ⋆ Juergen Gall ⋆ Peter Gehler ⋆ Christian Leistner ⋆ Andrea Fossati ⋆ Michael Breitenstein ⋆ Andreas Ess ⋆ Raphael Hoever ⋆ Thibaut Weise ⋆ Alain Lehmann ⋆ Daniel Roth ⋆ Bryn Lloyd ⋆ Till Quack ⋆ Angela Yao ⋆ Stefano Pellegrini ⋆ Jan Lesniak ⋆ Gabriele Fanelli ⋆ Nima Razavi ⋆ Marcin Eichner ⋆ Valeria De Luca ⋆ Lukas Bossard ⋆ Matthias Dantone ⋆ Stephan Gammeter ⋆ Dengxin Dai ⋆ Barbara Widmer ⋆ Christina Krueger ⋆ and all other current and former colleagues at BIWI.

Tatiana Tommasi ⋆ Joris Vangeneugden ⋆ Geert Willems ⋆ Michal Havlena ⋆ Alon Zweig ⋆ Jörg-Hendrik Bach ⋆ Hendrik Kaiser ⋆ Danilo Hollosi ⋆ all partners of the DIRAC EU-project ⋆ Michel Druey ⋆ Veronica Andrade ⋆ Nir Galili ⋆ Dominik Kamm ⋆ Raphael Eidenbenz ⋆ Michael Villiger.

Werner ⋆ Elisabeth ⋆ Silvan ⋆ Brenda.

Contents

1

Introduction

The automatic analysis of surveillance videos is a major field of investigation in computer vision research and industry. This holds especially for techniques that interpret the behavior in the monitored scene and is mainly due to the enormous variety of situations that occur in practice. One crucial aspect is to detect and report situations of special interest, in particular when unexpected things happen. Some solutions exist that work in well-constrained surveillance settings and show good results if they are tuned to a specific pre-defined application. However, for many application scenarios, an ideal automated visual surveillance system would autonomously interpret the scene and automatically recognize abnormal events. It would then notify operators or users accordingly, ideally including some semantic information with respect to the detected event.

In this thesis, the goal is to develop methods and techniques, that enable to approach such an ideal surveillance system. Within this active field of research, we propose techniques to automatically examine and interpret the behavior in the surveillance scene. A large part of the work is devoted to the visual analysis of human behavior and abnormal event detection in indoor scenarios. In this context, a working system would alert of dangerous situations and improve the personal safety of (elderly) people living on their own.

1.1 Video Surveillance

In the last two decades, the number of surveillance cameras installed to monitor private and public spaces has increased dramatically. This is mainly due to the rising fear of people about crime. To this end, cameras are installed in many public places, such as airports, train stations, city centers, or shopping

***Figure 1.1**: Traditionally, video streams from surveillance cameras are watched by human operators who are trained to spot abnormal events. An alternative is to record the video streams on storage devices for future viewing in case something interesting had happened. Recent approaches inspect the streams with rule-based processing. In this thesis, we investigate towards a fully automated analysis of the surveillance video for a robust detection of abnormal events.*

malls. Other interesting applications of visual surveillance systems include the examination of crowd motion, traffic flow monitoring, biometric identification, the assessment of industrial processes or human behavior interpretation in retail spaces. From an economic perspective, the visual surveillance market is huge, and a significant part is invested in adequate software solutions[1].

Deployed by companies such as IBM, Bosch, GE, Honeywell, Siemens, ObjectVideo, or BRS labs, current surveillance systems to some extent include automatic video processing. For example, techniques for the detection of an intruder, a car driving against the traffic or an unattended piece of luggage exist currently. They apply rule-based detectors that are manually tuned to well-defined settings, in order to raise an alert in case a suspicious configuration is met. An operator then has to verify the video stream and initiate the according actions as indicated in Figure 1.1. Hence, in general, real-time monitoring installations still rely on constant verification by a knowledgeable human operator. In contrast, many closed circuit television (CCTV) systems record the video to storage devices and delete it after a certain period. This is useful to go back in time and identify the involved persons, for example if a theft or an aggression had happened. Of course, this retrospective analysis does neither prevent the crime nor detects it when it happens. Consequently, and due to the

[1]The market research firm Markets&Markets estimate the global video surveillance market to grow to $37.7 billion in 2015 [Markets & Markets 2011].

***Figure 1.2**: Different visual surveillance scenarios at varying scale levels that we deal with in this thesis.*

large amount of human monitoring effort involved, smart surveillance software solutions are highly desirable [Frost & Sullivan 2008].

In computer vision research, sophisticated solutions to various surveillance tasks have been proposed in the last couple of years (see [Gong *et al.* 2011, Lavee *et al.* 2009, Hu *et al.* 2004, Dee and Velastin 2008] for surveys). The goal is to analyze the monitored scene and extract information that is useful for a previously specified application. Very roughly, the different tasks can be grouped into different scale levels, as illustrated in Figure 1.2 (see also [Breitenstein 2009]).

The visual processing for surveillance applications often starts with the tracking of foreground objects which are commonly obtained through background modeling [Stauffer and Grimson 1999, Cucchiara *et al.* 2003, Zhao and Nevatia 2004]. If the appearance of the tracked objects is known and can be trained a priori, tracking-by-detection approaches are a well-suited alternative, in particular if human behavior is examined (*e.g.*, [Andriluka *et al.* 2008, Breitenstein *et al.* 2009a, Wu and Nevatia 2007]). For an increased robustness, additional application-specific priors can be added, for example to model social interactions [Pellegrini *et al.* 2009] or include scene properties as in [Huang *et al.* 2008, Stalder *et al.* 2010]).

Tracked trajectories are then in many cases used to learn patterns of normal activity for scene-specific trajectory models (*e.g.*, [Stauffer and Grimson 2000, Hu *et al.* 2006, Basharat *et al.* 2008, Wang *et al.* 2006]) or to discover functional scene regions [Turek *et al.* 2010, Makris and Ellis 2005]. One

step further, [Hospedales *et al.* 2009, Kuettel *et al.* 2010, Wang *et al.* 2009] propose unsupervised techniques to mine spatial and temporal properties and rules that govern a traffic scene. They allow for data-driven segmentation and interpretation of a surveillance scene with respect to space and time. If larger scenes are monitored, target tracking sometimes fails due to very small, indistinguishable agents. In this case, flow and motion vectors can be computed and analyzed, for example in very crowded scenes [Ali and Shah 2007, Mehran *et al.* 2009].

If the analysis of human behavior in the scene needs to be more precise, a human centered perspective is required and detailed body motions are interpreted. Often, a set of predefined human actions are modeled with diverse techniques and recognized during runtime. Popular such approaches are described in [Efros *et al.* 2003, Gorelick *et al.* 2007, Laptev 2005, Dollar *et al.* 2005, Schindler and Van Gool 2008, Yao *et al.* 2010]. Not relying on labeled training data, other approaches automatically learn human action categories [Niebles *et al.* 2008] or temporally segment human activities [Zhou *et al.* 2008, Turaga *et al.* 2009] in an unsupervised manner.

Most of the mentioned techniques are specifically developed for the application in a single scale level (*c.f.* Figure 1.2) which limits their general applicability. In this thesis, we try to overcome this limitation.

1.2 Anomaly Detection

(a) Fall of a person *(b)* Shift change *(c)* Street festival

***Figure 1.3**: Abnormal events that occur in the different scenarios of Figure 1.2. All of them were detected with the techniques described in this thesis.*

Abnormal behavior detection is an important element for any surveillance system and often required in practice. This is true not only in visual surveillance,

but also for example for computer network security or to detect credit card fraud and illegal transactions in banking (see [Chandola *et al.* 2009] for a survey on anomaly detection). One major issue lies in the fact, that the application scenarios are very different and require the techniques to be adapted and tuned accordingly. In Figure 1.3 we show three exemplary abnormal events that were detected with the techniques proposed in this thesis.

In most computer vision approaches, abnormal events are identified as outliers to previously trained models of normality (*e.g.* [Johnson and Hogg 1995, Makris and Ellis 2005]). If motion patterns in the scene are available, statistical outliers are then detected as abnormal events based on low level motion features (*e.g.* [Adam *et al.* 2008, Stauffer and Grimson 2000]), or high-level specific object tracking [Hu *et al.* 2006, Basharat *et al.* 2008, Hendel *et al.* 2010]. In crowded scenes, optical flow vectors are quantized and modeled in order to detect deviations to normality, as for example in [Kim and Grauman 2009, Kratz and Nishino 2009]. Due to the fact that trained models of normality can hardly contain all expected configurations, Boiman and Irani [Boiman and Irani 2005] attempt to compose the current observation from space-time fragments in the database in order to detect irregular situations.

In an unsupervised approach, Zhong *et al.* [Zhong *et al.* 2004] learn a database of observed scene-motion prototypes and calculate the similarity in order to detect abnormal patterns. In the same spirit but to be more memory-efficient, Breitenstein *et al.*[Breitenstein *et al.* 2009b] apply clustering techniques to model normal behavior on a scene level. Using only appearance features, they can cope with the low, irregular frame-rate of a webcam stream. The beforementioned spatio-temporal scene modeling techniques of [Hospedales *et al.* 2009] and [Kuettel *et al.* 2010] also include the ability to report abnormal events as atypical configurations.

1.3 Indoor Monitoring of Humans

Of particular interest in this thesis is the application scenario of autonomous living where the detection of abnormal situation is of crucial interest. The daily life of mostly elderly persons is recorded and interpreted, for example in order to assess the health status of the person or to alert in case of suspicious situations. One approach to register the behavior in peoples houses is to install

many different sensors in so-called *smart homes* for technology-assisted living [Cook and Das 2007]. For example, Rachidi and Cook [Rashidi and Cook 2010] use motion and contact sensor recordings from several months to discover activities of daily living in an unsupervised manner. Introducing vision sensors, Zouba *et al.* [Zouba *et al.* 2009] fuse person tracking in videos and environmental sensory data at a high level to recognize daily living activities at home.

In the context of elderly person monitoring, fall detection is prevailing concern, as it represents a major health issue [Tinetti 2003]. Many solutions using various sensors have been proposed for the automatic generation of alarms in suspect cases [Noury *et al.* 2007]. The approaches include wearable, accelerometer based systems (*e.g.*, [Li *et al.* 2009, Chen *et al.* 2005]) but also concepts relying on vision. Wireless wearable devices are very reliable, however they have the clear disadvantage that the concerned person may forget to wear or to recharge them.

Vision based techniques on the other hand have the advantage of being non-invasive, as they monitor the person from a distance. Many of the approaches proposed in the past rely on precisely modeling a fall. During runtime, the event is re-detected and an alarm is emitted. Such approaches are for example based on the three-dimensional modeling of the visual hull of a person in [Anderson *et al.* 2009], the detection of a fall from shape and motion history [Rougier *et al.* 2007, Nasution and Emmanuel 2007], or rely on 3D head tracking [Rougier *et al.* 2006]. Nait-Charif and McKenna [Nait-Charif and McKenna 2004] use an overhead camera to track the person and define zones of usual inactivity. They argue that if the monitored person is inactive at a different locations, this corresponds to an abnormal event. In a different approach, Cucchiara *et al.* [Cucchiara *et al.* 2005] use a posture classification system for a more detailed human behavior analysis that permits the detection of a fallen person.

1.4 Our Paradigms

In contrast to the afore mentioned approaches for visual fall detection, we do not explicitly model a fall because we want to detect abnormal behavior in a larger sense. Unexpected motion, for example when the person is waving to signal something or when he suddenly starts to limp might also be of interest

and therefore should be detected by our algorithms. Furthermore, in contrast to most of the above mentioned approaches, we aim at the development of techniques which are broadly applicable. We will show how to monitor persons, but also apply the same techniques to different surveillance scenarios at different scale levels and reliably detect abnormal events. Due to these reasons, we follow the indirect path and model the observed behavior in the scene. Outliers and statistical deviations from this model of normality are interpreted as abnormal events.

Within the scope of developments and findings in the DIRAC research project[2] that are partly summarized in [Weinshall *et al.* 2012], we have initiated the usefulness of hierarchical models. This is especially true for the detection of abnormal events in surveillance scenarios. Due to the fact that we mostly do not know a priori what surprising situations to expect in the future, a reasoning at different levels of detail is called for. As we will show, subtle behavior changes, such a different person walking into the room can equally be detected as a major event, such as a fall. Additionally, the hierarchy paves the way for a semantic interpretation of the abnormal event.

One option is to learn such models of normal scene behavior from labeled training data. This will be demonstrated in Chapters 2 and 3 and has the advantage that a re-detection at runtime of the learned concepts is possible. In contrast, unsupervised model learning, as shown in Chapters 4 and 5, can easily adapt to the specificities of the observed behavior in a certain scene. Such models can include update mechanisms that account for shifts in the normal behavior during runtime.

Our long-term vision is to be able to mount a camera and process the video data without human intervention. The algorithms should autonomously pick-up the observed behavioral concepts, and adapt and refine the models of normality over time in order to detect any kind of abnormal situations. In this thesis, we make an attempt to move closer to this goal.

1.5 Contributions

The contributions of this thesis are summarized as follows:

[2]Project funded by the European Commission in FP7, IST-027787, "Detection and Identification of Rare Audio-visual Cues", www.diracproject.org.

- We apply multiple human body trackers on a single person. Every involved tracker follows a certain aspect of human motion, and they are arranged in a hierarchical, tree-like structure. From their relative interactions and confidence levels, we show how to detect different abnormal events. As the trackers are trained and labeled off-line, semantic reasoning becomes feasible.

- We show how to update a set of previously trained human motion trackers, by including a transfer learning stage. With minimal human intervention, the knowledge from the available trackers is used in order to label and train new activity trackers.

- In a different approach, we establish a data-driven approach to automatically model human behavior in hierarchies. This works without human intervention and includes separate models for human appearance and motion. We show how to detect abnormal events at runtime, that can be semantically interpreted in the hierarchies.

- In fact, this separate hierarchical description of (human) appearance and motion has biological relevance. We conducted a behavioral study, that showed very interesting results when comparing our approach to the behavioral responses of trained monkeys.

- The performance of the abnormal event detection is directly related to the quality of the established models of normality. We show that if we build such unsupervised models in a way that temporal constraints of activities are respected, meaningful human actions are automatically discovered in the modeling phase. In addition, when applied to unseen data, previous abnormal event detection techniques are outperformed.

- Not only (abnormal) human activities are considered, but we extend our techniques to multiple visual surveillance tasks. For example, we are able to interpret and monitor traffic scenes or observe various events in webcam streams from places of public interest.

- In industrial environments, the compliance with temporally consistent work-flows is crucial in many manufacturing processes. We show how to automatically extract such work-flow models and use them for the interpretation of abnormal events in industrial scenarios.

1.6 Organization of the Thesis

This thesis is structured as follows.

In Chapter 2, the idea of tracker-trees is presented. The reasoning among trackers that incorporate different assumptions is performed in a tree-like structure. To this end, we employ publicly available human trackers or detectors, but also present a silhouette-based tracking technique, that relies on low dimensional modeling of the activity. Experiments show the applicability in different indoor surveillance scenarios, where the behavior of a person is monitored. We show how the abnormal event detection in the tracker-tree outperforms other generative modeling techniques.

Chapter 3 extends the tracker-tree concept in order to learn and incorporate new activity concepts. We make use of recent transfer-learning techniques that act as an expert to label unknown activity samples. The augmented tracker-tree, obtained with minimal labeling effort then recognizes previously unknown activity concepts.

In Chapter 4, our first approach to unsupervised behavior learning and abnormal event detection is presented. In two hierarchies, responsible for appearance and motion encoding, respectively, the observations are modeled without any human interaction. At run-time, the two hierarchies are used to interpret new observations. If outliers persist, a concept change has occurred, and we additionally show how our model can be updated during runtime. We also demonstrate that the two hierarchies have in fact a biological motivation, as behavioral experiments performed with monkeys lead to corresponding results and interpretations.

In Chapter 5, we introduce a technique to unsupervised modeling of activities that makes use of the temporal structure of the human activities captured in the video. On one hand, this leads to activity models that turn out to be well interpretable and can perform state-of-the-art automatic action discovery. On the other hand, if these models are used for the interpretation of unseen videos, activities are re-detected and abnormal activities are spotted robustly and accurately.

Chapter 6 aims at extending the concept of abnormal behavior spotting to other, not person-focused scenarios. We show applications for road / traffic monitoring and the surveillance of public spaces from webcam streams. To this end,

we need to introduce adequate feature representations. Finally, we modify the previous method and extend it with additional assumptions to build a representative work-flow model. This enables the automatic discovery and monitoring of industrial work-flows.

Chapter 7 finally concludes this dissertation, discusses the important findings and gives lines of future research.

2

Tracker-Trees

2.1 Introduction

In this chapter we detect abnormal events following the indirect route of detecting them as deviations from models of usual human behavior. This said, in order to cover the wide range of unusual events that may be of interest, this calls for modeling a wide spectrum of usual events, often at different levels of granularity. We do this in a supervised setting, where different concepts of normal behavior are known and integrated in the model. For instance, the calamity of falling would be detected as deviating from all the normal categories like walking, standing, or sitting. Our proposal is to build an entire tree of trackers, as sketched in Figure 2.1. The aspiration is to detect a gamut of unusual events, which will also gradually get more subtle and semantically rich as one moves up in the tree.

The idea of fusing multiple trackers as parallel or cascaded observers is not new and has for example successfully been used for visual tracking problems. Toyama and Hager [Toyama and Hager 1999] introduced the concept of *Incremental focus of attention* that uses multiple trackers at different levels of accuracy for robust tracking. They argue that more robust trackers on lower levels can be used as fall-back options and for re-initialization if more precise trackers loose the target due to unexpected visual perturbations. Whereas the tracker configuration is fixed in this work, Stenger *et al.* [Stenger *et al.* 2009] learn how to combine multiple observation models for target tracking. In parallel or cascaded evaluation, they switch between the observers based on the observer's confidence, in order to robustly determine the target location over a long time.

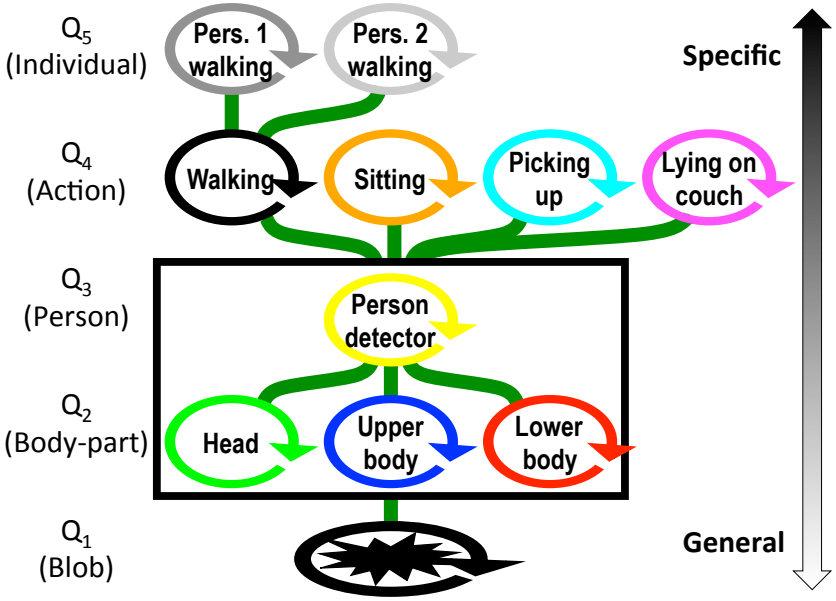


***Figure 2.1**: Overview of the proposed tracker tree with increasingly informed trackers for increasing concept levels Q_1 to Q_5. Each circle depicts one implemented tracker. On level Q_1, any kind of foreground blob is tracked. Level Q_2 encodes partial human appearances, whereas on level Q_3 entire persons are detected. Level Q_4 involves trackers that go after specific human actions, and on level Q_5, action trackers are tuned to individual persons.*

Whereas the underlying idea of these works is somewhat similar in the sense that they rely on more robust and more specific trackers, we follow a very different goal. Robust target tracking is not our principal rationale to count on such trackers. In the tracker-tree, we want to detect expected and unexpected configurations of trackers with high tracking confidence. Therefore we explicitly construct a hierarchy of trackers, with possibly multiple trackers at the same level. From the simultaneous outputs of a multitude of trackers, we show how to detect normal and abnormal behavior in the scene. Since these trackers all follow a known concept and are arranged in a hierarchical manner, a semantic interpretation can be deduced on the nature of the abnormal event.

In this chapter, we introduce the idea of tracker-trees in detail in Section 2.2. As we want to apply a tracker-tree in an indoor monitoring scenario, we need trackers that are suitable for the application. Therefore, we explain in Section 2.3 our method for appearance-based activity tracking, that we use for many trackers in the tracker-tree. Subsequently, we show our implementation of the tracker-tree for abnormal human behavior detection (Section 2.4). In Sections 2.5 and 2.6, an extensive number of qualitative and quantitative experiments is proposed.

The main parts of this chapter were published in [Nater *et al.* 2009].

2.2 Tracker-Tree Concept

Visual trackers in general incorporate a certain amount of information about the normal situations they are applied to. For example, an articulated body motion tracker is highly tuned to a walking person and exploits strong priors for successful tracking, whereas a simple blob tracker relies on very weak assumptions. We propose to arrange multiple different trackers in a tree-like hierarchy, where the location of each tracker is based on the information it relies on. Trackers further up in the tree have been trained for a narrow activity concept whereas trackers closer to the root node are able to track a broad variety of motion patterns. The *tracker-tree* described here is geared toward the detection of unusual events in the home (*e.g.*, for elderly care) where we assume the video camera to be static. The principle however is not restricted to this scenario. The used tracker-tree is shown in Figure 2.1, where the circles correspond to the implemented trackers. The green interconnections indicate the hierarchical dependencies.

The root node on level Q_1 is a simple, generic blob tracker, going after anything moving that is not background. One level up, inside the black box ($Q2$ and $Q3$), a tracking-by-detection framework is used to track multiple body parts (lower body, upper body, head-shoulders) at level $Q2$ and to detect a person at level $Q3$. On these levels, people are tracked, independent of their activity. One further level up on Q_4, four trackers detect a person walking, sitting, picking up an object or lying down. This level could be considered an action-specific level. Then there is one higher level (Q_5), which specializes the walking tracker towards trackers that are tuned particularly towards the gait of specific people. Hence, our hierarchy consists of multiple levels, within which

families of trackers are trained to cover the normal conditions at that level. Notice that at one level there could be multiple such families. For example, if one would also have a running tracker at the action level, it would make sense to also have a family of person specific running trackers one level up. Going to higher levels, the trackers are endowed with stronger and stronger knowledge about the (normal) world. In our current implementation, all the different trackers operate autonomously, *i.e.* a tracker does not depend on the outcome of any other.

Unusual events are detected when and where a level can deal well with an event (can explain it with the available trackers), whereas none of the relevant trackers at the immediately higher level can. This is motivated by the fact that a tracker that uses more knowledge about the world should be more robust. If none of the more informed trackers can deal with the data, but the less informed one can, then this is a sign that something unusual is going on. Indeed, using more information is only advantageous as long as this information is correct. In the case of an unusual event, none of the usual, extra pieces of information apply. A performance reversal occurs in the sense that the weaker tracker better explains the data than any of the more informed trackers. An interesting aspect of the hierarchical approach is that unusual events at multiple, semantic levels can be handled and interpreted. For instance, if none of the people trackers can explain the data well, but the blob tracker follows an object, we may have a pet entering the home of a person not having one. If none of the normal action specific trackers does well, but tracking by full body detection still works, this might be an indication of an unusual human action like limping. If walking is detected, but the gait does not correspond to any of the known individuals, an intruder or at least someone not observed before seems to be in the house. As elderly people often are the victims of scams, this would indeed be noteworthy and a sufficient condition to activate some remote attention by an assistant.

To explain the concept more formally we re-use the terminology of [Weinshall *et al.* 2012], in the sense that we find in our tracker-tree simultaneously *disjunctive* and *conjunctive* nodes. Each tracker instance is characterized with a confidence score q, that captures how well it can interpret the observed data. This score can equally be binary.

As a disjunctive example within the tree, a person (general level) can perform different actions, which are modeled by different action trackers (specific level; here walking, sitting, picking up and lying down). Thus, the total score q_4 of the more specific concept on level Q_4 is

$$\mathsf{q}_4 = \mathsf{q}_{\mathrm{Walk}} + \mathsf{q}_{\mathrm{Sit}} + \mathsf{q}_{\mathrm{Pickup}} + \mathsf{q}_{\mathrm{Liedown}}. \tag{2.1}$$

An *incongruent* or abnormal motion pattern occurs if a discrepancy exists between the more general and the more specific classifier, *i.e.*

$$\mathsf{q}_{general} \gg \mathsf{q}_{specific}, \quad \text{here}: \ \mathsf{q}_3 \gg \mathsf{q}_4 \tag{2.2}$$

In other words, an abnormal activity is reported if a person is in the scene (Q_3 active) but non of the known activities on level Q_4 is detected.

In the same vein, individual walking trackers on Q_5 are sub-concepts to the generic walking tracker, and model the gait pattern of individual persons known to the system.

On the other hand, the trackers inside the black box in Figure 2.1 form a conjunctive hierarchy that considers the person as composed of body parts. Separate detectors check whether legs, upper body, and body shoulder patterns are found. In case the person is fully visible and in a familiar pose, all three parts should be detected. From the conjunctive perspective, the indication strength of finding a person amounts to

$$\mathsf{q}_2 = \mathsf{q}_{\mathrm{Head}} \cdot \mathsf{q}_{\mathrm{UpperBody}} \cdot \mathsf{q}_{\mathrm{LowerBody}} \tag{2.3}$$

In this case, an incongruent event is detected if $\mathsf{q}_2 \gg \mathsf{q}_3$, *i.e.* all body parts are detected, but no person is tracked in the scene. In fact, we found little practical use for this way of handling conjunctive tree sections, but we propose an adapted reasoning for occlusion handling.

Occlusion Handling Partial occlusions occur frequently in in-house surveillance scenarios, *e.g.*, furniture partially blocking the view of a person. In the tracker tree, this means that a person is detected and q_3 is valid, but at least one of the body parts fails (q_2 small). Without proper training (including training images of occlued objects), such an occlusion leads to the case that $\mathsf{q}_2 \ll \mathsf{q}_3$ which according to [Weinshall *et al.* 2012] theoretically corresponds to an invalid model.

We propose a different interpretation which considers the body part trackers as conditioners for the action trackers. Since the actions (level Q_4) are trained on examples of fully visible persons, the validity of any of these trackers cannot be

expected to hold when the person is only partly visible. In the case of occlusion by a sofa, for example, the lower body part is missing and therefore no action is expected to be valid, as all action-specific trackers are critically dependent on the visibility of what are in that case the relevant body parts, *i.e.* the legs.

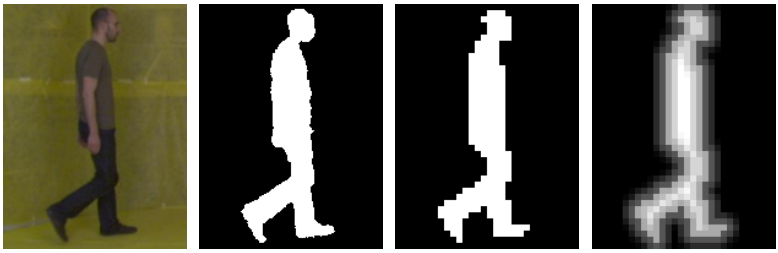
To address this problem, occlusion detection is incorporated into the approach and prevents trackers that need invisible parts to flag incongruences. For example, if a person is observed (yellow detector) but not all of his body parts, this blocks incongruences higher up in the tree from being signaled if the absence of that body part precludes action detectors from functioning properly.

2.3 Appearance-based Probabilistic Activity Tracking

In this section, we describe a shape based tracking approach which is based on manifold learning and nicely fits in the concept of more or less informed trackers. This technique allows for the creation of different trackers by combining different sets of training data and consequently making the trackers more or less specific. We use this family of trackers at different levels in our tracker-tree implementation.

Manifold learning is a popular technique in human activity modeling and recognition (see [Moeslund *et al.* 2006] for a survey on human motion understanding). The fact that consistent human actions have a small number of intrinsic degrees of freedom can be exploited for designing a low dimensional manifold which describes the principal aspects of the observed human activity while omitting details. Learning manifolds and mapping functions to appearance space and body pose space is for example used successfully to infer 3D body poses from silhouettes [Elgammal and Lee 2004, Jaeggli *et al.* 2007], possibly including dynamical information as in [Urtasun *et al.* 2006a, Li *et al.* 2007]. In a Bayesian tracking framework, the human motion with its dynamics is encoded in low dimensional manifolds that are used to estimate observation likelihoods (*e.g.* [Lee and Elgammal 2007, Gammeter *et al.* 2008]).

Inspired by these ideas, we establish our tracking method which unlike other approaches does not infer 3D human poses, but simply models and interprets the person's appearances in a probabilistic form. While more sophisticated



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

***Figure 2.2**: Image representation: (a) original, (b) segmented, (c) rescaled, (d) distance transformed.*

descriptions of human body configurations could be used, we argue that for the detection of abnormal situations in our indoor setting, the analysis of human silhouettes appears to be adequate

Representation. In order to encode the shape of the tracked persons, we use silhouettes obtained from background subtraction. Inspired by [Zivkovic and van der Heijden 2006], the background is adapted over time, and foreground masks are obtained in YUV color-space. The silhouettes are resized to a fixed dimension and normalized, as shown in Figure 2.2. The binary images are then further encoded by a signed distance transform, bounded to maximal and minimal values and finally each frame is reshaped in a vector.

2.3.1 Model Generation

In the training phase, we generate a model representing the human appearances from a training video with one action concept. The extracted feature vectors are stacked into a feature matrix.

Dimensionality reduction. The high dimensionality of the shape representation space is reduced with respect to the included training data. As we want to use the low dimensional manifold for tracking, we impose the requirement that the chronological order of the input frames has to be reproduced in the embedding. This means, that the Euclidean distances of consecutive frames, measured in latent space, should be small.

We use Isomap [Tenenbaum *et al.* 2000] as a nonlinear dimensionality reduction technique, which has proven to meet our expectations and also produces compact and interpretable manifolds. We fix the latent space to be three dimensional, encoding enough variance for successful reconstruction and still permitting efficient tracking. An example of such manifold is shown in Figure 2.3. It was obtained from one person's continuous unconstrained walking. Points correspond to video frames and their temporal order is indicated by the connections. For some of the frames, the according silhouette is displayed. Gray-scales qualitatively reflect the dimension which intuitively encodes the persons walking direction, reaching from right (light grey) to left (dark grey), with frontal/dorsal orientations in between. The manifold also represents the person's gait with open leg states being spatially separated from closed ones.

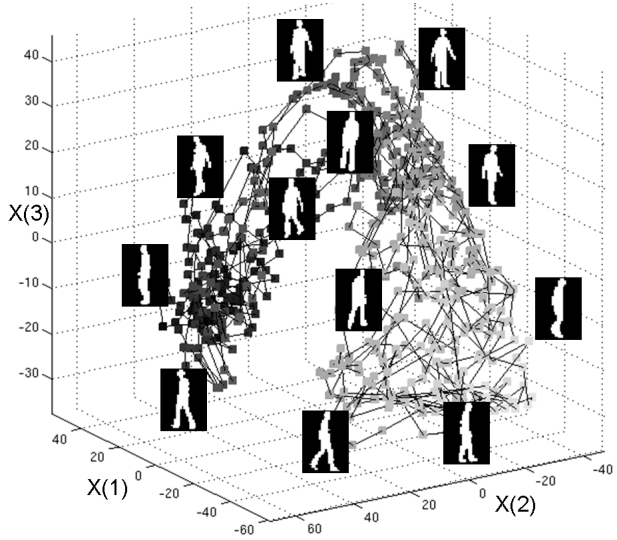
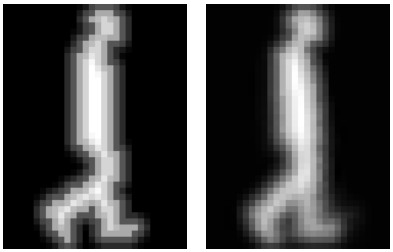


***Figure 2.3**: Visualization of the low-dimensional representation: Manifold of 600 encoded silhouettes obtained from a sequence of one person's unconstrained walking (See text for details).*

Gaussian Process regression. Isomap is a data-driven technique to reduce the dimensionality of the input data based on local distances, therefore no explicit mapping is formulated in the framework (in contrast to linear dimensionality reduction techniques such as principal component analysis). For the genera-

tive association between latent space and image representation, we learn a regression function by using Gaussian Process [Rasmussen and Williams 2006]. To this end, we employ the Gaussian Process toolbox [Lawrence 2003] and compute the mapping function $\mathcal{M} : \mathbf{z} \mapsto \mathbf{y}$ which estimates the shape representation $\mathbf{y}$ and its variance for any latent point $\mathbf{z}$. For the ease of notation, we denote the predicted shape obtained by mapping a predicted latent point $\hat{\mathbf{z}}$ simply as $\hat{\mathbf{y}}(\hat{\mathbf{z}})$. In Figure 2.4(b), an feature example in image space representation is shown after Isomap embedding and Gaussian Process reconstruction of Figure 2.4(a).



(a) *(b)*

***Figure 2.4**: Illustration of silhouette modelling in the manifold: (a) Input descriptor, (b) embedded and reconstructed.*

2.3.2 Tracking

After having learned a low dimensional manifold representing a set of encoded silhouettes for a specific action class, the next step is to explain unseen test sequences within this model. This is done with a Bayesian tracking approach [Doucet *et al.* 2000] by using a six dimensional particle filtering technique. For every hypothesized sample $\theta_i = \{u_i, v_i, s_i, \mathbf{z}_i\}$ the observation likelihood is evaluated, where $\{u, v\}$ is the bounding box location in the image, s its scale (with fixed aspect ratio), and $\mathbf{z}$ the tracked shape in the low-dimensional embedding space[1].

[1]The six dimensions in the search space are determined from the three dimensions in image space and the dimensionality of the underlying embedding. In our experiments, we used three-dimensional embeddings, that showed to reflect well the trained actions. If actions of a different complexity would be modelled, the particle filter would need to be adapted accordingly.

Likelihood formulation: For every particle θ_i, the likelihood of the shape observation given this sample is estimated, using the following formulation:

$$\mathbf{p}\left(\mathbf{y_{obs}}|\theta_i\right) \propto \mathcal{N}\left\{d\left(\mathbf{y_{obs}}, \hat{\mathbf{y}}(\theta_i)\right); 0, \sigma^2\right\} \tag{2.4}$$

with $d\left(\mathbf{y_{obs}}, \hat{\mathbf{y}}\right)$ a distance function between the observed $\mathbf{y_{obs}}$ and the predicted $\hat{\mathbf{y}}$ shapes, both represented as distance transformed silhouettes. The likelihood is in this case normally distributed with zero mean and σ^2 variance. More precisely, if we denote the shapes $\mathbf{y_{obs}}$ and $\hat{\mathbf{y}}$ as vectors of K elements y_k and $\hat{y}_k$, $k = 1 \ldots K$, the distance function becomes

$$d\left(\mathbf{y_{obs}}, \hat{\mathbf{y}}\right) = \sum_{k=1}^{K} \beta_k |y_k \hat{y}_k| \tag{2.5}$$

with

$$\beta_k = \begin{cases} 1 & \text{if} \quad \text{sign}(y_k) \neq \text{sign}(\hat{y}_k) \\ 0 & \text{if} \quad \text{sign}(y_k) = \text{sign}(\hat{y}_k) \end{cases} \tag{2.6}$$

such that equally signed pixels in the observed and the predicted shape do not increase the distance.

With this likelihood formulation we obtain a posterior probability density function over all samples θ_i given the shape observation $\mathbf{y_{obs}}$ for each frame in the test sequence.

Illustration: The proposed silhouette based tracking approach is illustrated in Figure 2.5, where two frames of a publicly available video sequence[2] are shown. On the upper left of each frame, the background subtracted silhouette is presented, and on the lower left, the image space representation of the particle filter sample with the highest weight is shown. The latter corresponds to the shape encoded in the low dimensional model which best matches the observed silhouette. Trained in a controlled lab setup, the learned model can nonetheless be applied on any sequence and accurate tracking is possible even with noisy background subtraction. The tracking approach works well as long as the observed shape can be well described by the model, *i.e.*the likelihood term has

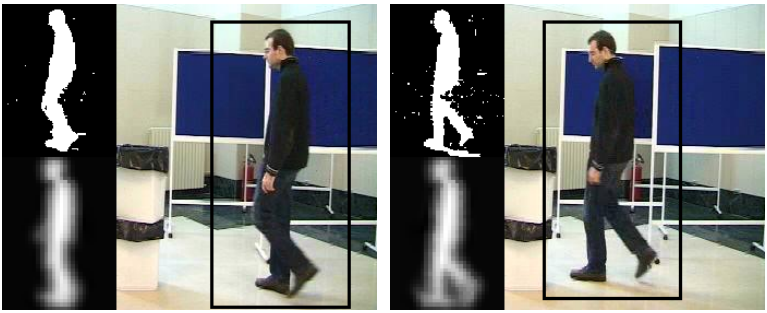


***Figure 2.5**: Application of the proposed silhouette based tracking approach on a publicly available video sequence, from which two frames are shown. The background subtracted image and the best corresponding shape in the model are depicted on the left.*

clearly pronounced peaks, whereas it results in small posterior probabilities for out-of-model observations.

Tracking priors: In this probabilistic formulation it is possible to easily incorporate scene-specific knowledge by adding tracking priors. For example, in some experiments a ground plane prior is used for the stabilization of the tracker and for limiting the search space of the particle filter.

2.4 Implementation

The outline of the proposed tracker-tree was already presented in Figure 2.1, where all the implemented trackers were schematically depicted in circles. The tracker instances are placed on five hierarchical concept-levels $Q_1 - Q_5$. Next, each level will be detailed and the corresponding trackers that we employ on this level are explained. We use previously proposed state-of-the-art methods as well as the custom-built technique of Section 2.3 trained for different levels of generalization.

[2] video downloaded from www.openvisor.org, 2009/05/27

2.4.1 Trackers in the Tree

Level Q_1: The least informed and thus most general tracker in the *tracker tree* is meant to trace any foreground blob, subject to any kind of deformation. This is a simple, generic blob tracker which uses no information about the nature of the foreground object. For this purpose, a color-histogram based CAMShift tracking approach [Bradski 1998] is used in its OpenCV implementation. This tracker delivers an ellipse of the approximate target location in each frame.

Level Q_2: The trackers on this level make a first step towards the description of human body shapes. For tracking people independent of their activity, we use a set of body-part trackers, namely for the lower body, the upper body and the head-shoulders. For these three trackers, we generate an embedding using the method of Section 2.3. The image data provided during the training procedure is chosen with respect to the specified body-part of the tracker. For each of these three trackers, the obtained low-dimensional manifolds are similar to the one shown in Figure 2.3 and encode the principal motion such that tracking remains possible within this manifold. Particle filter based tracking is accomplished and the output is a probability that quantizes the match between observation and body part model.

Level Q_3: The tracker instance on level Q_3 in the upper part of the black box in Figure 2.1 is a state-of-the-art person detector based on discriminatively trained part models [Felzenszwalb *et al.* 2008]. It is used as provided by the authors on the website and follows a tracking-by-detection approach.

Level Q_4: On this level, we are interested in tracking different basic human actions. As shown in Figure 2.1, we dispose of four different action trackers. Each of them is based on training data modeled as described in Section 2.3. The trained action concepts are unconstrained indoor walking, sitting on a chair or on the couch, picking up an item from the floor and lying down on a couch. For generalization reasons, these trackers are trained from recordings of multiple people performing the actions. Consequently, the manifolds encode some variability with respect to execution style of the action. For example, the walking manifold looks similar but denser and richer compared to the one in Figure 2.3 which was obtained from a single person's walking only.

Level Q_5: On the most specific level in the current tracker-tree, the aim is to track one particular person performing a specific action. In that sense, this

is a specialization of the action specific trackers one level down and we rely therefore on a modification of the non-personal walking tracker in Q_4. The goal is here to separately model the appearance of two persons by providing individual training data and learning two distinct manifolds. Besides tracking the considered person, the tracker outputs a probability quantizing how well the observed silhouette fits in with the individual model. This output score is evaluated relative to the non-personal walking tracker at level Q_4: The discrepancy in terms of posterior probability between person specific and non-personal trackers provides information on the walker's identity.

2.4.2 Setup

Training. The trackers that are based on the technique of Section 2.3 require a training with data corresponding to the activity concept each tracker reflects. This training is done in a supervised manner with accordingly segmented training data. To this end, we record a set of training videos in a controlled environment, using an RGB camera with a VGA frame resolution and a frame rate set to 15 fps. The extracted full-body silhouettes are resized to 40×40 pixels, for the body-parts, this full-body representation is separated into smaller parts. For the trackers in Q_2 and Q_4, recordings of $3-5$ persons are used. Each of these trackers is trained with $1000-3000$ images. The two person-specific trackers in Q_5 are trained with recordings from single persons, each of them comprising approximately 500 frames. The noise term σ^2 in the particle filter (*c.f.* Equation 2.4) is estimated from the training data.

Runtime Analysis. At runtime, when applied to unseen video sequences, we run all the trackers in parallel. Initialization on the first frame in image space is done either manually, or with indication from the person detector in Q_3. In manifold space, the particles are initialized randomly. As seen in Section 2.2, each tracker needs to deliver a confidence score q, such that reasoning in the tree can be performed. This confidence score reveals whether the tracker is confident in explaining the current observation or not. For the Bayesian trackers in Q_2, Q_4 and Q_5, the respective scores are set to be the (temporally smoothed) data likelihood of Equation 2.4 corresponding to the particle with the maximal weight. The total number of particles is empirically fixed to 1500. The blob tracker in Q_1 is always active as long as the person is in the scene and the per-

son detector (Q_3) is said active if a person is detected. Hence, q_1 and q_3 are binary.

2.5 Illustrative Experiments

We illustrate the functioning of our tracker tree in a series of experiments and demonstrate its capacities for the detection of abnormal situations in living-room scenarios. The four subsections here depict short illustrative use cases, each of them highlighting a different aspect of the tracker-tree. Subsequently, in Section 2.6, we demonstrate the operation of the tracker-tree in a more comprehensive and quantitative manner[3].

2.5.1 Illustration 1: Normal Operation

In a first experiment, we want to show the properties of the different trackers in the *tracker tree* applied to a scene without any abnormality. In Figure 2.6, an extract of a publicly available sequence[4] is given, showing a person entering the room, walking a couple of steps and sitting down. Three snapshots from this video are displayed in Figure 2.6(a-c). For the entire sequence, the tracker's output probabilities on a logarithmic scale are plotted in the bottom part of Figure 2.6 and the instants corresponding to the frames are indicated by vertical black lines. In the images, the white ellipse indicates the general object tracker [Bradski 1998], the other bounding boxes correspond to the trackers. The color code of Figure 2.1 is applied. In the probability graph we introduce an empirically determined threshold which is used to decide on the reliability of the tracker. In other words, this threshold could indicate to the system whether the particular tracker is likely to explain the observation. The threshold is indicated by a black dotted horizontal line and accordingly, only trackers with above-threshold probabilities are visualized in the frames. Note that the yellow bounding box in the images corresponds to the part model detector [Felzenszwalb *et al.* 2008], for which no probability output is available and thus only a binary curve is plotted. In this sequence however, this detector is always active.

[3] Most videos can be downloaded from www.vision.ee.ethz.ch/fnater/tracker-trees/.

[4] video downloaded from www.openvisor.org, 2009/05/27

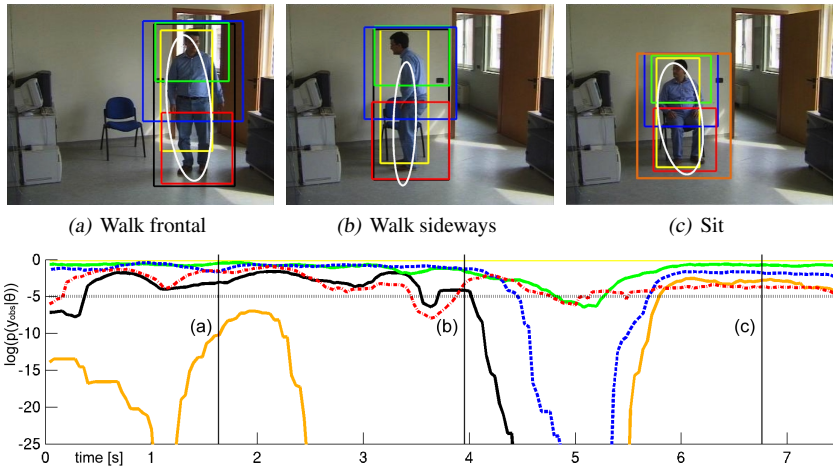
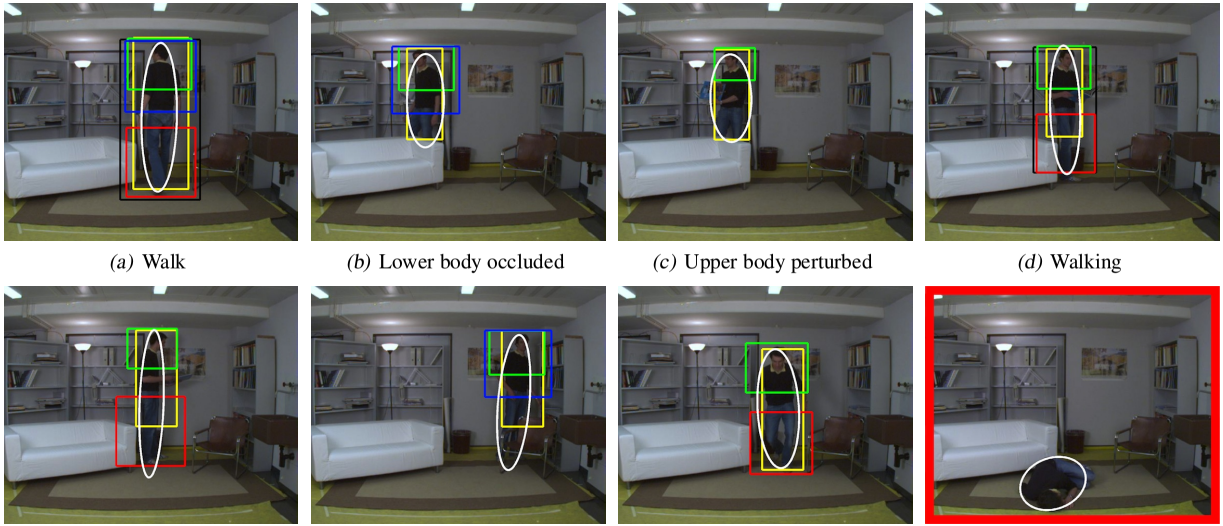


***Figure 2.6**: Results for the first test sequence: The behavior of the different trackers in the framework is presented. The images on top show three frames from the sequence, their corresponding instants are indicated in the plot on the lower part. The confident trackers are visualized in the frames. The color code is taken from Figure 2.1 (see text for interpretation).*

From the lower part of Figure 2.6 it can be seen that as long as the person is walking (a,b), the observation is well explained by the underlying model of the walking tracker (black bounding box and line) as the output probability is high. When the person starts to sit down the walking tracker fails and also the lower body part tracker has a transitory instability. Thereafter, the person remains seated (c) and the sitting tracker is able to explain the situation. All the body-part trackers are also active. This small example shows the basic functioning of the tracker-tree, applied to test data that was recorded in a very different setting (different person, different scene), compared to the training setup.

2.5.2 Illustration 2: Occlusion Reasoning and Fall Detection

In a second sequence presented in Figure 2.7, a person is in the room (a), walking behind the sofa towards the shelf on the left (b), taking a book and reading it (c), turning towards the other shelf (d, e), where the book is placed.



(a) Walk *(b)* Lower body occluded *(c)* Upper body perturbed *(d)* Walking

(e) Upper body perturbed *(f)* Lower body occluded *(g)* Stumbling *(h)* Fall

***Figure 2.7**: Results for the second test sequence: The system's reaction in the case of occlusions and an abnormal event is illustrated. When the person is reading a large book, the upper-body model is perturbed. Only the confident trackers are displayed with bounding-boxes.*

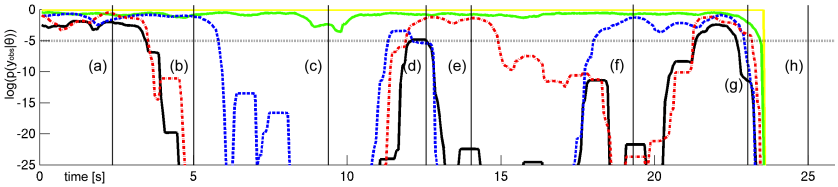


Figure 2.8: *Evolution over time of the tracker confidences for the sequence of Figure 2.7. The color code of Figure 2.1 is used and the person-specific walking trackers are omitted.*

Coming from behind the chair on the right (f), he wants to move to the front, when suddenly he stumbles across the edge of the carpet (g) and falls (h). The same color code is used as in the previous video, omitting the unused sitting, picking and lying trackers as well as the person-specific walking trackers.

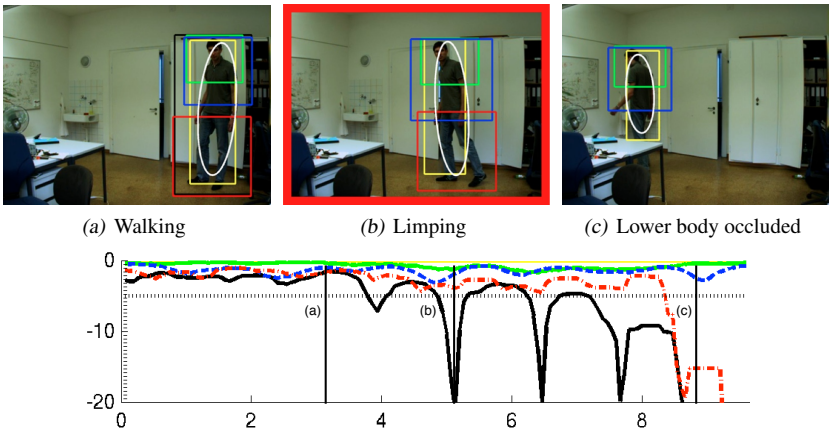
Again, the relative tracker outputs allow for interpretations of what is going on in the scene. The evolution of tracker confidences over time is displayed in Figure 2.8. For example, the system detects an occlusion (b,c,f) if the walking tracker fails, but some body parts, in this case head-shoulder and upper body are still visible. In the same category falls the event in (e), where the person is holding a book such that the upper body tracker is perturbed. This demonstrates the use of body part trackers, especially in living room scenarios where multiple occluders are usually present.

A fall is detected when a foreground object is tracked but cannot be explained by any of the more specific trackers, as seen in Fig 2.7(h). None of the tracking models trained for normal human behavior can cope with this special situation. Here we additionally make use of the sequential information that we had observed a person right before the fall happened.

2.5.3 Illustration 3: Limping

In a third experiment, we show the tracker tree's behavior to a person in the scene who is limping. This action was not included during the training phase and should therefore be detected as unusual. In a short sequence shown in Figure 2.9, a person is walking normally towards the right, turns around (a)

and suddenly starts to limp (b). Finally, his legs are occluded by the desk in the foreground (c). From the bottom part of Figure 2.9, where the same signals are plotted as in the previous experiments, it can be seen that all the trackers behave normally until approximately second 3, when the person is walking normally. When he starts to limp, all the body part trackers (on level Q_2 in the implemented tracker tree) and the person detector in Q_3 are still following the target, the walking tracker in Q_4 however shows periodic drops in its output score. This occurs in the part of the walking cycle, where the limping is characterized, *i.e.* when tightening the leg. In this case, an abnormal event is noted from the fact that all lower level trackers agree, whereas no higher level tracker explains the situation. In the end of the sequence, where the legs are invisible, no evidence is given for abnormal walking, due to the fact that not all body part trackers remain active.

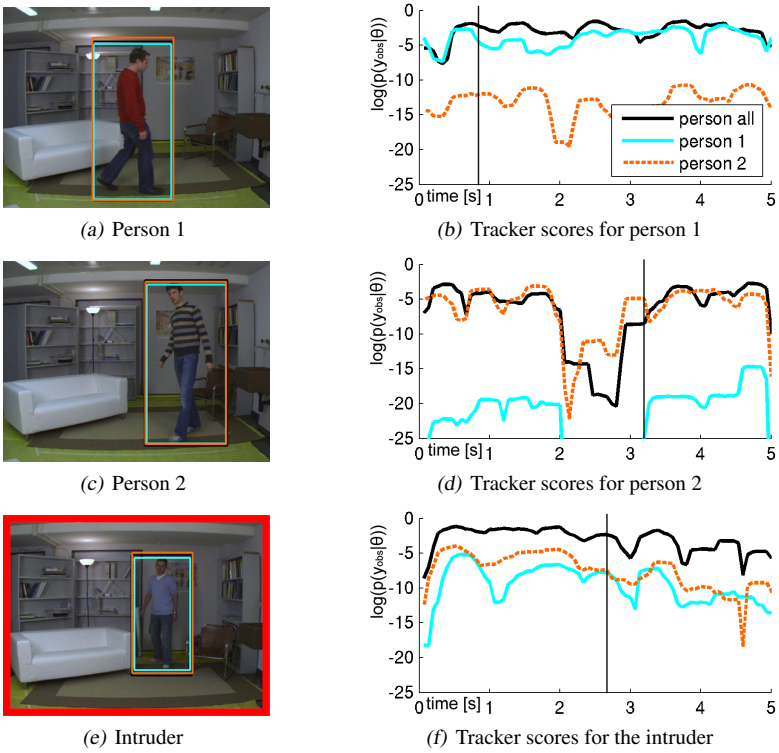


(a) Walking *(b)* Limping *(c)* Lower body occluded

***Figure 2.9**: Illustration of the tracker trees output to a limping action. The abnormal event is detected temporarily during the walking cycle, all part trackers agree whereas the (normal) walking tracker cannot cope with the situation.*

2.5.4 Illustration 4: Intruder Detection

For the intruder detection task, we include the two person specific walking trackers (Figure 2.1, level Q_5) as well as the general person walking tracker in Q_4. In Figure 2.10, the principle is demonstrated with three short sequences,



(a) Person 1 *(b)* Tracker scores for person 1

(c) Person 2 *(d)* Tracker scores for person 2

(e) Intruder *(f)* Tracker scores for the intruder

***Figure 2.10**: Results for the person identification task. Three different persons are walking through the room, and the corresponding tracker output scores are plotted aside. From the discrepancy between general (black) and the two person specific models (cyan and orange), the abnormal walking pattern belonging to an intruder is spotted. For visibility, all the other trackers in the system are omitted. The vertical black line in the plots denotes the instant for which the video frame is displayed.*

starring two familiar and one unknown person respectively. In Figure 2.10(a,b), we track the first familiar person. In this case, the probability outputs for the multiple person tracker (black) and the person 1 tracker (cyan) correlate, while the person 2 tracker (orange) explains the situation less accurately. The opposite happens in Figure 2.10(c,d), where the second known person is tracked. If a third, unknown person is in the scene (Figure 2.10(e,f)), his appearance is

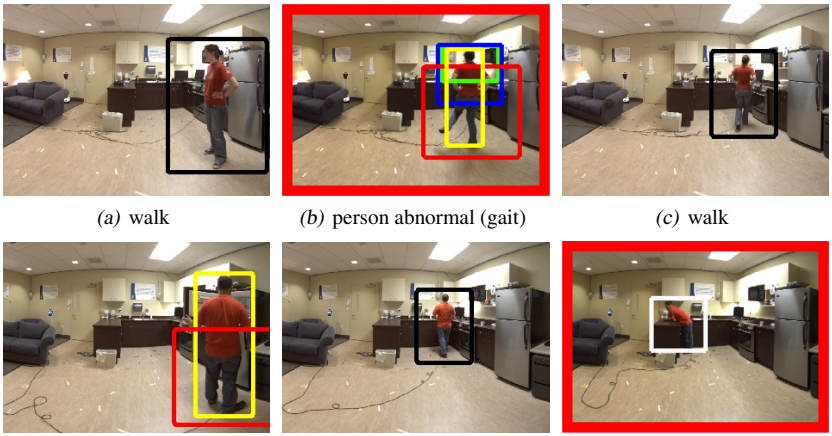
well modeled by the general person tracker, while the person specific ones tend to fail. The person must therefore be an intruder (in the sense of someone not known to the system yet).

2.5.5 Illustration 5: Recordings in a Different Setup

In the DIRAC project, we have received video data from the Living Lab at Oregon Health and Science University (OHSU). The data was recorded with a RGB-camera in RAW format at 24 fps and different persons acting in a small apartment. For an increased field of view, a fish-eye lens was employed. For our tests, we use two videos and show example results on short extracts.

To run the tracker-tree, the videos need to be preprocessed, *i.e.* the frames are debayered, the fisheye-distortion is corrected [Havlena *et al.* 2009] and the images are cropped. Then, the videos are analyzed in the tracker tree and abnormal situations are reported. As the recording setup and the person's behavior differs quite considerably from the training conditions, it is also interesting to highlight some failure cases. In Figure 2.11, a few frames of the two sequences are displayed. The indicated trackers are again the active trackers at the highest level in the tree and the color-code of Figure 2.1 is used.

In the first row, normal walking and an abnormal gait pattern are correctly detected. In fact, in Figure 2.11(b) the actress moves her leg over the audio cable in an unexpected manner in order not to stumble. This can be seen as an abnormal gait pattern, similar to the limping case. In the second row of Figure 2.11, the trackers are distorted due to erroneous foreground extraction (opening the refrigerator in (d)). The new foreground object perturbs the head-shoulders and the upper-body trackers. In Figure 2.11(f), the person leans over the kitchen desk. This motion is apparently unfamiliar to all trained concepts. Since the blob tracker still tracks the person, an abnormal event is signaled in the same way as for a fall.



(a) walk *(b)* person abnormal (gait) *(c)* walk

(d) person incomplete (?) *(e)* walk *(f)* object abnormal (?)

***Figure 2.11**: Per-frame results for the OHSU video sequence. In (d) and (f), the tracker-tree brakes down and fails to correctly interpret the situation.*

2.6 Quantitative Experiments

2.6.1 Experiment 1: ETHZ Sequence

We perform a more in-depth evaluation of the tracker-tree on a video sequence which was recorded in a living-room environment. A single person is monitored and incongruent events are spotted. The test video of about 1000 images contains diverse every-day actions such as walking, walking behind occluding objects, sitting on different chairs, or picking up small objects. It also contains abnormal events, *e.g.*, when the person falls, limps, jumps over the sofa or when an intruder enters the room.

In Figure 2.12 we present the output scores of the different trackers for a short fragment of the video sequence. The plotted curves depict the confidences of the individual trackers, the color code of Figure 2.1 is applied. The horizontal line indicates the threshold that is used to assess if a tracker is confident or not. The reasoning in the tree is then performed and the detected incongruent events are highlighted in red.

In Figure 2.13 we show exemplary result frames, where the active trackers are visualized as bounding boxes in corresponding colors. As long as the person

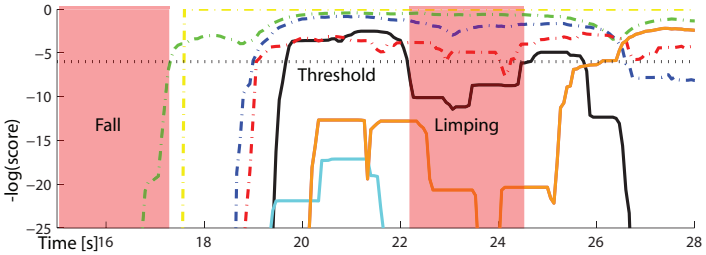
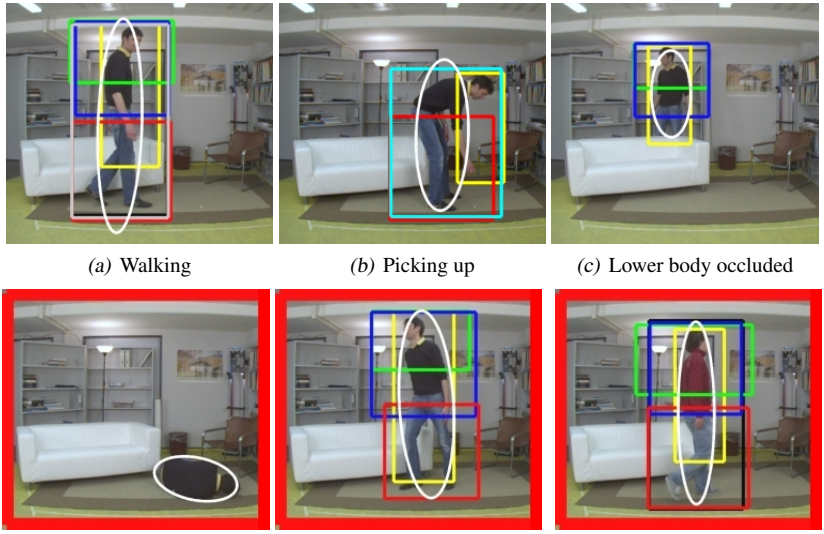


Figure 2.12: *Extract of one tracked sequence, the tracker output scores are plotted over time, the color code of Figure 2.1 is used. The individual walking trackers are omitted and the horizontal threshold is used for classification. For illustration, incongruent patterns are highlighted in red.*



(a) Walking *(b)* Picking up *(c)* Lower body occluded

(d) Fall *(e)* Limping *(f)* Intruder

Figure 2.13: *Selected frames from one sequence. The active trackers are visualized by the bounding box using the color code of Figure 2.1. If an incongruence occurs, the entire frame is marked in red.*

behaves according to expectations (walks, picks up an object, or walks behind the sofa), the tracker-tree accepts the situation. When an incongruence in

the motion pattern is detected, an abnormal event is detected and the frame is marked in red (fall, limping, intruder).

In the following, we analyze the performance of the tracker-tree for abnormal event detection and compare it to state-of-the-art methods. To this end, we sweep the threshold that is applied to the tracker confidence scores and compare the tree's output with the ground truth annotation of the test sequence. As baseline comparison, we learn a Gaussian Mixture Model (GMM) from the training data using the EM algorithm [Bishop 2007, McLachlan and Krishnan 1997]. Similarly to most of our trackers, GMMs represent the data in a generative manner, but without a hierarchical structure. We train the k mixtures with the same full-body silhouette representations of the different actions that were used for training the trackers on level Q_4.

The results are displayed as ROC curve in Figure 2.14. Note that the ROC curve for the tracker-tree has a particular shape and does not reach full recognition since the nonlinear classifier reasoning is applied after fixing the threshold. Due to the reasoning in the hierarchy, the tracker-tree outperforms GMM outlier detection regardless of the number of mixture components.

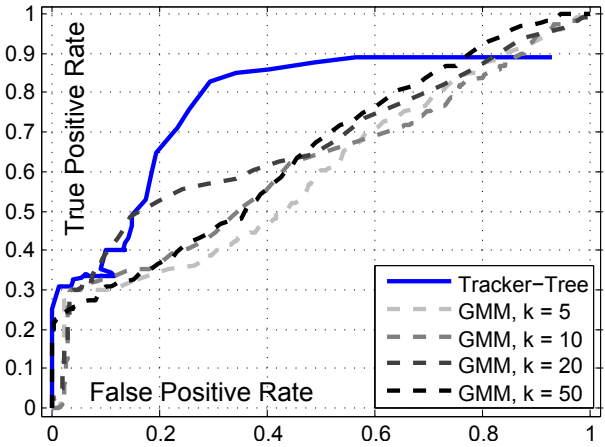


***Figure 2.14**: ROC curve evaluation of abnormal action detection using the tracker-tree. Due to the hierarchical reasoning, tracker-trees outperform comparable state-of-the-art methods based on GMMs.*

2.6.2 Experiment 2: DIRAC Data

To demonstrate the use of the tracker-tree in the DIRAC project, we installed a mock-up living room at the Computer Vision Lab at ETHZ. The scene consists of a couch, a chair, a television, a shelf, cupboards, a carpet on the floor and a lamp in the background. This installation is similar to the previous one, that was used to record training data and for previous experiments. The video was recorded with an AVT Marlin fire-wire camera, having a resolution of 640x480 pixels and a frame rate of 15 frames per second.

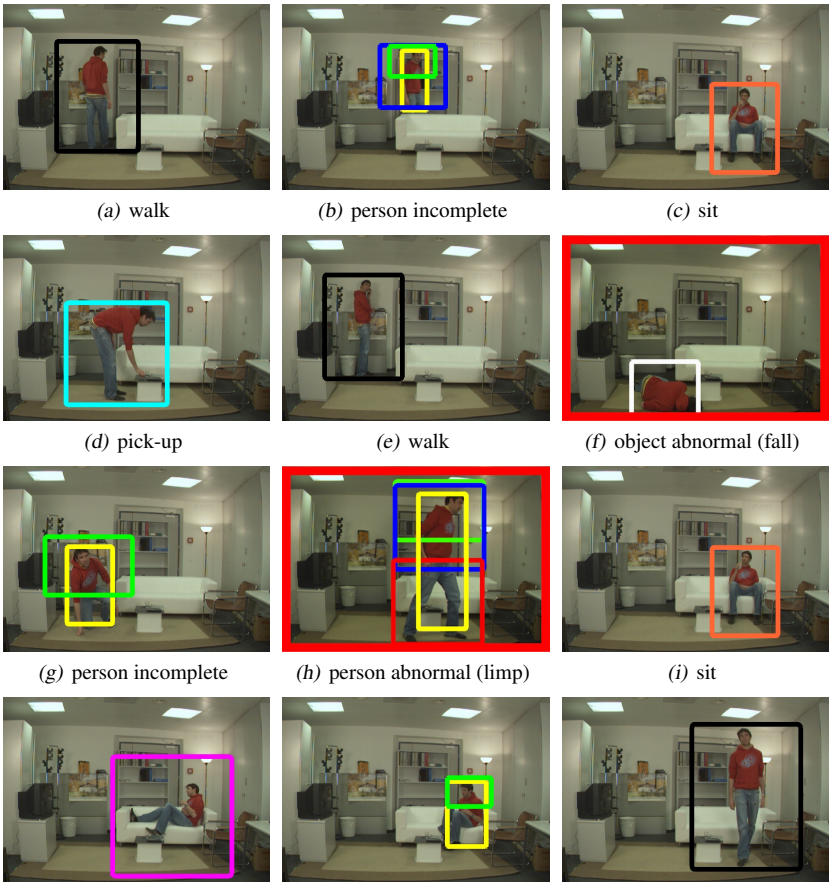
In this test video, the person walks into the living room and performs various actions: walks around, walks behind the couch, sits down, reads, stands up, falls, lies down, etc. The test video contains 2380 frames.

Qualitative experiments

In Figure 2.15 we show a number of selected frames for the DIRAC sequence. The person is tracked throughout the video sequence, and the active trackers are displayed. For the sake of visibility, not all active trackers are shown, but always the most informed (highest in the tree of Figure 2.1) and most confident is depicted, using the color-code of Figure 2.1. For example, if walking is observed, only the bounding box output of the walking tracker is displayed in black, even though other less informed trackers, such as the foreground blob tracker, or the different body part trackers, are also active.

The person is tracked precisely in terms of location and size of the bounding box throughout the entire sequence. Different trackers appear at different instances in time, according to the activities performed by the protagonist. We now run through a number or interesting cases:

When the person moves behind the couch, the lower body is not visible from the camera location, hence the person is half occluded and not all body parts are tracked. This can be observed in Figure 2.15(b). As said previously, this corresponds to a missing part inside the black box of Figure 2.1. Since this situation might happen often in our scenario, we do not signal an anomaly, but rather observe an 'incomplete' person. In this case we do not expect any higher-level tracker to be active, since they have been trained on entirely visible persons. Not detecting all body parts however may also have other reasons. If for example the person has some unknown upper or lower body pose (as in



(a) walk *(b)* person incomplete *(c)* sit

(d) pick-up *(e)* walk *(f)* object abnormal (fall)

(g) person incomplete *(h)* person abnormal (limp) *(i)* sit

(j) lie down *(k)* person incomplete *(l)* walk

***Figure 2.15**: Selected frames of the DIRAC sequence. The active trackers on the highest level in the tree is displayed, using the color code of Figure 2.1 The detected abnormal events are indicated with a red frame.*

Figure 2.15(g)), the head might still be tracked (green), and the person is still detected by the person detector (yellow). This actually happens often during transitions from one action to another. These transitions might not be modeled by the action trackers, but body parts are observed. An example is the transition from lying to sitting, as depicted in Figure 2.15(k). Other actions, such as

picking up an object from the floor, or in our case picking a pen from the table are recognized as such (Figure 2.15(d)). Also, lying down on the couch is successfully tracked (Figure 2.15(j)).

If the person falls and lies immobile on the floor, none of the trackers that go after normal human motion or body parts in normal configuration remain active, and only a foreground object is tracked (white bounding box in Figure 2.15(f)). In this case, an abnormal event is reported. After standing up, the person might have hurt himself and starts to limp (Figure 2.15(h)). In this case, the walking tracker, trained on normal walking motion will lose confidence, while all the body part trackers still validate the observation. In this case, an anomaly is detected. As all the body parts are visible, this must be an unknown action.

Quantitative evaluation

As we have already demonstrated the abnormal event detection capacities of the tracker-tree in the previous subsection, we here follow a different aim. We want to quantify the action detection performance of the action trackers in the tree. Hence, we evaluate the manifold-based action trackers on level Q_4 in Figure 2.1 individually and jointly in terms of recall and precision.

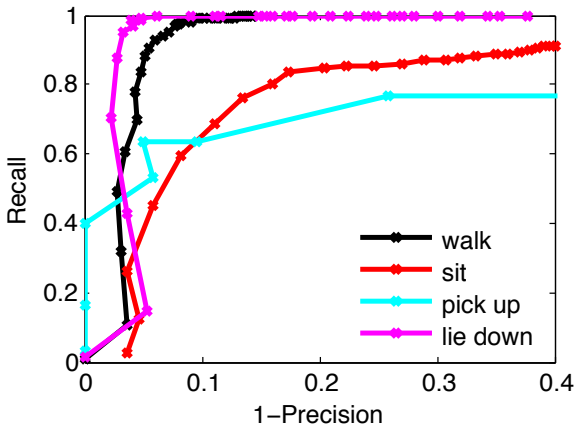


***Figure 2.16**: Evaluation of the action recognition performances of our activity trackers. Each trackers is run individually and its score is compared to the ground-truth.*

Figure 2.16 depicts the four Recall-Precision-Curves for the action trackers, (walk, sit, pickup and lie down, respectively). The curves are obtained individually, which means that each tracker assesses the observation independent of the output of the other trackers. We sweep the detection threshold that is applied to the tracker output score and compare it to the manual ground-truth annotation[5]. From these curves, it appears that the walking and lying trackers are relatively precise, whereas the picking-up and sitting trackers perform worse.

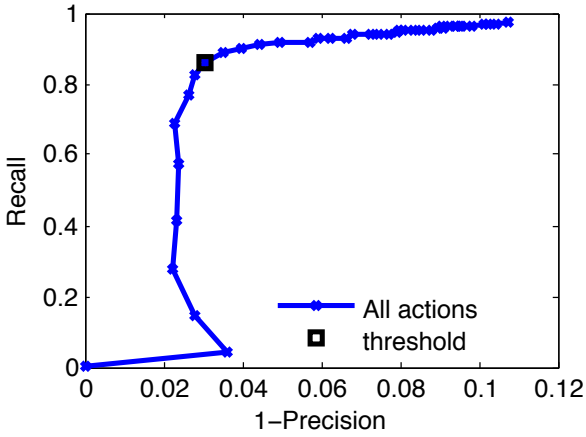


***Figure 2.17**: Overall performance of the action level Q_4 in the tracker-tree. The max-pooling operation improves the action detection performance. The indicated threshold is used for the tracker display in Figure 2.15.*

The RPC depicted in Figure 2.17 combines the outputs of all the trackers. In fact, from the conjunctive tracker hierarchy, we know that only one tracker at this level in the hierarchy can possibly be active at the time. Therefore a max-pooling operation is performed before applying the classification threshold. This improves the correct detection of the performed action (note the different scale on the abscissa). In black we indicate the threshold that is chosen for the per frame results in Figure 2.15. At this threshold, the action trackers show a recall of 86% at a precision of 97%.

[5]The *recall* is the true-positive rate and quantifies the retrieved positive samples (*i.e.* detected abnormal frames) compared to the ground-truth of positive samples. The *precision* is the fraction of the correctly detected positive samples with respect to all detected samples. For better readability $1 - \text{precision}$ is shown.

To evaluate the abnormal event detection capacity of the tracker tree, we apply the threshold of Figure 2.17 on the tracker's scores. Then the abnormality reasoning is performed in the tracker-tree and we compare the per-frame output to the manually annotated abnormality labels. The overall system has a recall of 75% at a precision of 81% for abnormality detection task on the DIRAC sequence.

2.6.3 Discussion

We see from the experiments that it is useful to interpret the independent tracker outputs in a tree-like structure. Irregularities with respect to the expected output scores of the different trackers can be detected quite reliably. A reasoning is possible since all the trackers have previously determined tracking capacities and target classes. Depending on the tree level, wehre the anomaly is signalled, different interpretations can be given. This is important, as the appropriate response will depend in the type of alarm. Furthermore, the higher the quality of the available trackers and the preciser they are tuned to the application scenario, the better the events in the scene can be interpreted.

The major limitation of the current tracker-tree implementation is certainly the silhouette-based tracking technique of Section 2.3. Such trackers will fail under circumstances that make background subtraction intractable, such as illumination changes, displaced furniture, moving camera or high noise level. In particular, the intruder detection relies on shape characteristics of the individual persons, and is highly sensible to noise. Other more sophisticated techniques (*e.g.*, gait recognition) could be used instead for improving the performance. However, these issues concern the current implementation of the tracker-tree, but do not affect the validity of the presented anomaly detection concept. On the positive side it should be mentioned that in all the observed failure cases, the tree never completely broke down, and that the trackers recovered automatically, for example when the refrigerator in Figure 2.11(d) was closed again.

2.7 Conclusions

In this chapter, we have proposed tracker-trees as a model of human behavior. Unusual events are detected in cases where the behavior is not modeled explicitly. The underlying idea is to build a hierarchy of trackers in a supervised

and pre-defined manner. The location of each tracker in the global structure depends on how strong its prior expectations about the world are that are being exploited. We have argued that the multi-layer hierarchy allows one to make rather specific interpretations about the detected activity and the kind of unusual event that has occurred. High up in the tree semantically more complicated events are detected than lower down. From qualitative and quantitative examples, we show the applicability of the tracker-tree for the detection of abnormal events, such as falls, in indoor surveillance scenes, where (elderly) people are being monitored when they are alone.

It is clear that unusual is not the same as important. Not all unusual events detected by such system will be relevant, and vice versa. An extensive such system will contain explicit trackers and detectors for several relevant cases. To make a step towards this direction, we present in the next chapter a technique to update the tracker-tree with minimal manual effort.

Clearly, the tracker tree as proposed here is just an example of how such structures may be put to use. A practical system would require the integration of many different trackers, selected with respect to the target application.

3

Activity Update via Transfer Learning

3.1 Introduction

In order to recognize the (abnormal) activities of a person in an in-house scenario, we have proposed the tracker-tree in the previous chapter. We have shown how known concepts can be recognized and labeled, and abnormal events are detected. Specialized trackers are trained with manually segmented and labeled training videos, as outlined in Section 2.3. However, this training process is cumbersome and limits the adaptation to new activity concepts or to different application settings.

We propose to augment this static model with an update procedure, based on transfer learning. To classify the activity samples, that are unknown in the tracker tree, we build a multi-class model which exploits prior knowledge of known classes and incrementally learns the new actions. This transfer-learning stage requires minimal human effort and provides labels for the new activities.

We give an overview of the update procedure in Section 3.2, the transfer learning technique is summarized in Section 3.3 and experimentally validated in Section 3.4.

The contents of this chapter are based on [Nater *et al.* 2011c]. The work was done in a close collaboration with Tatiana Tommasi and Barbara Caputo from IDIAP Research Institute in Martigny, Switzerland.

3.2 Overview

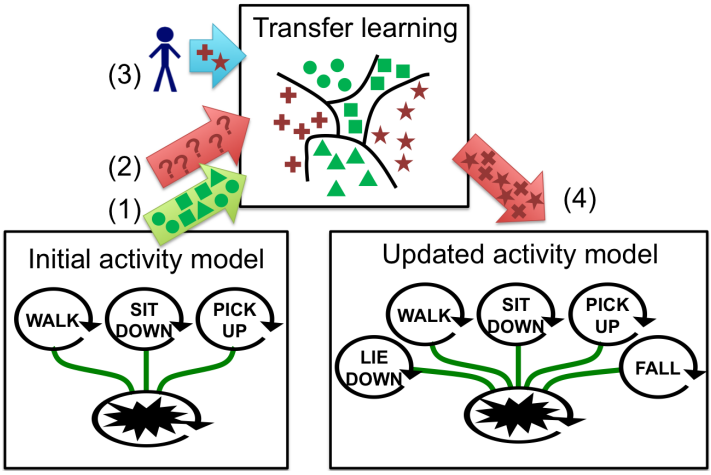


***Figure 3.1**: Schematic overview of our approach to combine activity tracking with transfer learning. In surveillance videos, an initial model recognizes familiar activities (1) or detects abnormalities (2). Together with minimal human interaction (3), the transfer learning algorithm returns labels (4) such that the activity model can be extended with new classes.*

The overall procedure of the proposed activity update is outlined in Figure 3.1. We follow the previous tracker-tree concept, but reduce the tree to two levels, *i.e.* we focus in particular on level Q_4 of Figure 2.1, where each tracker is trained to one specific activity class and also include the blob tracker of Q_1. The trackers are run in parallel. A user-defined threshold, applied on the activity trackers' output scores, determines active and inactive trackers. Of all the active trackers, the one with the maximal posterior probability determines the activity label of the current frame and the bounding box of the person, as illustrated in Figure 3.2. The cropped and labeled frames are delivered to the transfer learning stage (Arrow 1 in Figure 3.1). If non of the known action is observed, the foreground-blob tracker determines the bounding box of the person, which is handed over, labeled as unknown (Arrow 2). In the transfer learning stage, the information from minimal human annotation (Arrow 3) and the familiar action concepts is exploited to label these unknown samples (Arrow 4). From these newly labeled samples, we learn new manifold-based

activity trackers with the technique of Section 2.3, and they are integrated besides the existing ones. In this sense, the transfer learning algorithm acts as an artificial expert to label previously unlabeled samples.

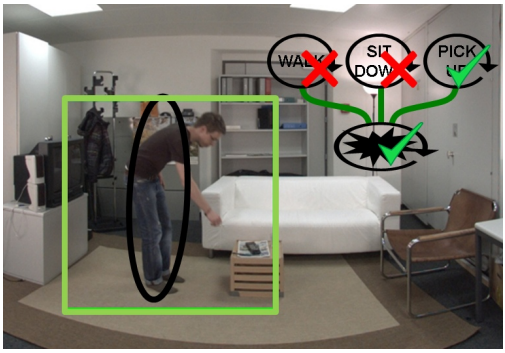


Figure 3.2: *To demonstrate the update of the tracker-tree, we partially use the original tree of Figure 2.1. The person is tracked by a set of activity-specific trackers and the general foreground blob tracker. If an action tracker is active (here: picking up in green), it provides labeled bounding boxes (Arrow* 1 *in Figure 3.1).*

The interaction between activity tracking and transfer learning is useful due to their complementary nature:

- Generative tracking with multiple activity trackers provides labels for familiar activities and detects abnormal situations. In both cases, the location of the person is determined with a bounding box.
- Discriminative classification interprets the abnormal situations in order to label new activities. Knowledge transfer uses prior information from known classes for a more efficient and accurate labeling of new ones. Human annotation of at least one frame is necessary to provide the desired semantic label.

The approach has several application-specific advantages. Firstly, if only few labeled samples of some actions are available, we can exploit prior knowledge acquired under different conditions in terms of location, observed person and employed recording camera. Furthermore, human annotation of one sample

per class enables the semantic interpretation of the activities. For example, it is desirable to include a fall in the model, in order to automatically take appropriate action in case it is detected again, *e.g.*, call an ambulance. Like this, the model continuously becomes richer in what it knows, such that diverse activity concepts can be recognized and the performance increases over time. Finally, a shift in an activity concept, *e.g.,* a person gradually starts to limp, can also be integrated.

Related Work. The use of transfer learning for activity recognition problems has been introduced in recent works for example for cross view action recognition [Liu *et al.* 2011], for domain adaptation [Xian-Ming and Shao-Zi 2009, Yang *et al.* 2010, Hu *et al.* 2011] or to transfer across sensor networks [van Kasteren *et al.* 2010]. Furthermore, in a scenario similar to ours but not using video data, Rashidi and Cook [Rashidi and Cook 2011] use transfer learning to adapt models of daily activities between different residents in different smart homes. However none of these works consider the possibility to update the set of class knowledge models when the newly acquired information contains actions which were not seen before. In visual object classification, knowledge transfer is applied to solve a N' class problem when N classes are already known, with N' and N disjoint groups [Lampert *et al.* 2009, Jie *et al.* 2011].

3.3 Knowledge Transfer for Unusual Event Learning

Transfer learning can help to reduce the labeling effort for recognizing a set of new activities. The idea is to transfer from known classes the useful part of information while solving the new multi-class problem. Here, we briefly describe the employed method, as proposed in [Tommasi *et al.* 2010], and extend it to a multi-class setting.

Adaptive knowledge transfer

From a set of l labeled samples $\{\mathbf{x}_i, y_i\}_{i=1}^{l}$, where $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$ and $y_i \in \mathcal{Y} = \{-1, 1\}$, the goal is to learn a linear function

$$f(\mathbf{x}) = \mathbf{w} \cdot \phi(\mathbf{x}) + b \tag{3.1}$$

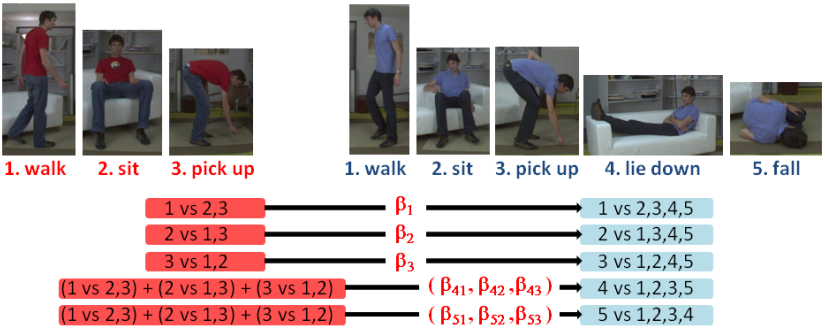


***Figure 3.3**: Schematic presentation of the transfer learning strategy. The activity classes on the left (red) are prior knowledge, the right classes (blue) are the new target activities. In a multi-class one-vs-all scheme new hyperplanes are obtained. For classes 1,2 and 3 we learn from the corresponding source knowledge while for classes 4 and 5 a weighted combination of all the known hyperplanes is used as prior. (Figure credits: Tatiana Tommasi)*

which assigns the correct label to an unseen test sample $\mathbf{x}$. The function $\phi(\mathbf{x})$ maps the input samples to a high dimensional feature space where the inner product can be easily calculated through a kernel function [Cristianini and Shawe-Taylor 2000]

$$K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x}) \cdot \phi(\mathbf{x}'). \tag{3.2}$$

As opposed to the classical theory of Least-Square Support Vector Machines (LS-SVM) [Suykens *et al.* 2002], the optimization problem is slightly modified by introducing a regularization term, that accounts for the adaptation to classes of prior knowledge [Tommasi *et al.* 2010]. The idea is to constrain the new model to be close to a set of k pre-trained models. With a linear dependency with respect to the models of prior knowledge $\mathbf{w}'_\mathbf{j}$ (*i.e.* $\beta_j \mathbf{w}'_\mathbf{j}$), the objective becomes:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w} - \sum_{j=1}^{k} \beta_j \mathbf{w}'_\mathbf{j}\|^2 + \frac{C}{2} \sum_{i=1}^{l} \zeta_i [y_i - \mathbf{w} \cdot \phi(\mathbf{x}_i) - b]^2. \tag{3.3}$$

In this case, $\mathbf{w}$ and b are the initial model parameters and ζ_i a weight introduced to cope with unbalanced data distributions, For a solution of this objective function, the Leave-One-Out (LOO) prediction is used as an unbiased estimator of

the classifier generalization error and hence, to find the best value of β. For the detailed solution, we refer to [Tommasi *et al.* 2010].

One-vs-All multi-class extension

Let's start from a prior knowledge problem with N different activity classes and train a multi-class SVM classifier with the one-vs-all approach. Only the parameters that describe the hyperplanes $\{\mathbf{w}'_\mathbf{j}\}_{j=1}^{N}$ are memorized while the data is not stored. As target task we consider to solve a $(N + N')$ multi-class problem where N categories are the same as in the original source task and N' classes are new. However, now only very few samples for each class are available.

The binary transfer approach described previously can be used separately to learn each of the $(N + N')$ one-vs-all hyperplanes (see Figure 3.3). The N hyperplanes associated to the same classes considered in prior knowledge, are now trained to separate some new positive samples against a different negative set due to the presence of N' new classes. In these cases the $\boldsymbol{\beta}$ vector reduces to one single value in $[0, 1]$. The method also exploits a linear combination of prior knowledge hyperplanes to separate each of the N' new categories from all the others. The idea that a combination of visual characteristics, which differentiate between the known actions, can still be useful to characterize the new ones when only few labeled samples are available.

3.4 Experiments

We perform two experiments to demonstrate the activity update. First, we show the performance of activity classification via transfer learning, then we verify how the newly learned classes improve the activity recognition performance of the (reduced) tracker-tree model. We use the same data for both tasks.

3.4.1 Dataset and Setting

In our experiments, we include 5 different activities to be recognized. These are *walk, sit down, pick up, lie down* and *fall*. We consider different cases that

might also appear in real-life scenarios. As depicted in Figure 3.4, we include two different indoor scenes, two camera types that were used for recording and three different persons.

(a) Camera 1, Scene 1 *(b)* Camera 1, Scene 2

(c) Camera 2, Scene 1 *(d)* Persons 1, 2, 3

***Figure 3.4**: Different settings are used for the experiments. We recorded in two different indoor scenes, with two different cameras and three persons perform the activities.*

Cameras. Camera 1 has VGA resolution and records at 15 frames per second. The used lens introduces minimal distortion. Camera 2 has a resolution of 1624×1234 pixels and records at 12 frames per second. A fish-eye lens with a large field of view introduces distortion, that needs to be corrected. To this end, we apply the technique of [Havlena *et al.* 2009] and rectify the images cylindrically, *i.e.* straight, physically vertical lines are preserved. For visualization purposes, the relevant image region is cropped out in Figure 3.4(c).

Sequences. We dispose of 12 video sequences, which were recorded as detailed in Table 3.1[1]. They contain between 1000 and 3000 frames and depict

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

a single person who performs all the five activities. We manually provide a frame by frame ground truth annotation for each sequence. Transitions (*e.g.,* standing up after a fall) are termed with *no activity*.

Seq $1a$, Seq $1b$, Seq$1c$: {Scene 1, Person 1, Camera 2}
Seq $2a$, Seq $2b$, Seq$2c$: {Scene 1, Person 2, Camera 2}
Seq $3a$, Seq $3b$, Seq$3c$: {Scene 1, Person 3, Camera 1}
Seq $4a$, Seq $4b$, Seq$4c$: {Scene 2, Person 3, Camera 1}

Table 3.1*: Three sequences were recorded for every parameter combination.*

Initial processing. We run the tracker tree with the three initial activity trackers (walk, sit down, pick up) and the blob tracker on all the sequences. The known activities are spotted and abnormal events are detected. Each frame is labeled and the bounding box of the person is obtained. This forms the basis for further analysis.

3.4.2 Transfer Learning

As explained in Section 3.2, the transfer learning step is used as an expert exploiting prior knowledge and labeling new samples that are then used to update the tracking system. Having an accurate classification process is crucial for the performance of the final action recognition method. We validate the proposed transfer approach in four cases. As prior knowledge we used Seq $*a$ with the $N = 3$ activities labeled in the initial processing. Seq $*b$ is used to extract randomly 10 frames for each of all the $N + N' = 5$ actions (initial processing and new activities). This defines the training set for the target task. Finally Seq $*c$ is used as test set.

The PHOG features [Bosch *et al.* 2007] (histogram bins=9, angle=180, levels=3) are calculated on the provided bounding box around the person and they are used together with the RBF kernel in all the experiments. The learning parameters are chosen by cross validation on prior knowledge. To implement the multi-class transfer learning method we started from [Tommasi *et al.* 2010] using the code released by the authors[2].

We compare three methods that are applied to the test sequence:

[2] http://www.idiap.ch/~ttommasi/

- *Initial Model*: The prior knowledge model learned on the 3 initial activities.
- *No Transfer*: The model learned on few samples of the 5 activities.
- *Transfer*: The model learned on few samples of the 5 activities transferring from prior knowledge.

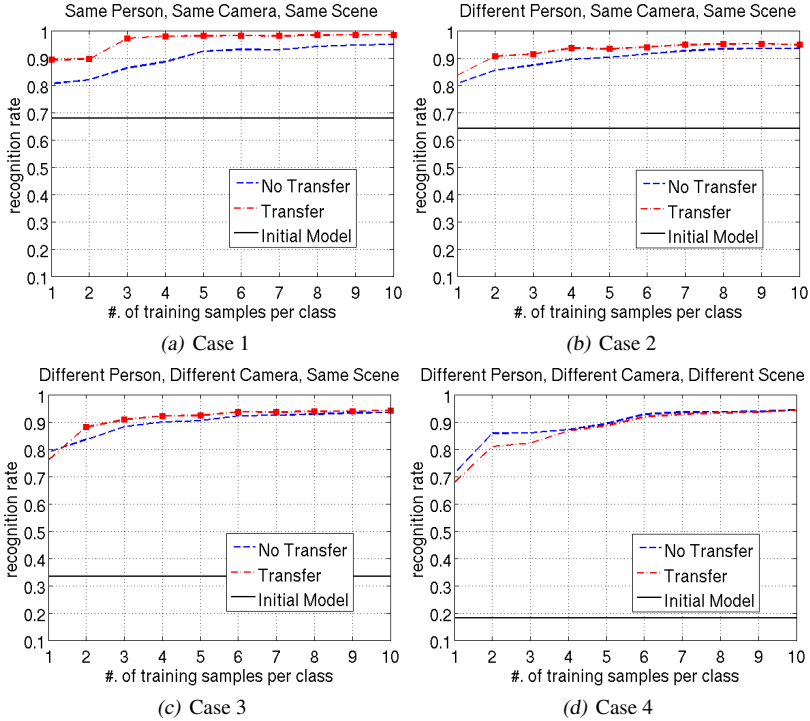
The plotted values correspond to the average recognition rate on 10 runs of the experiment (the random selection of training frames from Seq $*b$ is repeated 10 times). The significance of the comparison between *Transfer* and *No Transfer* is evaluated through the sign test [Gibbons 1985]: a square marker is reported on the graph if $p < 0.05$. The four experiments differ by the existing relation between prior knowledge and target task and the results are presented in Figure 3.5.

Case 1: same person, same camera, same scene. The acting person, the background scene and the recording camera are the same in prior and new sequence. Specifically we used Seq $1a$, Seq $1b$ and Seq $1c$. Classification results are reported in Figure 3.5 (a): transferring from prior knowledge guarantees a significant advantage compared to learning from scratch. The same experiment was repeated using Seq $3a$, Seq $3b$ and Seq $3c$, with equal results.

Case 2: different person, same camera, same scene. The background scene and the recording camera are fixed, but the acting person in prior knowledge is different with respect to the one in the training and test videos. We used respectively Seq $2a$, Seq $1b$ and Seq $1c$. The results are reported in Figure 3.5 (b). Even if the actions in prior knowledge are performed by a different person, transferring information still guarantees an advantage in learning. The same experiment was repeated inverting the role of the two acting persons and using Seq $1a$, Seq $2b$ and Seq $2c$ with analogous results.

Case 3: different person, different camera, same scene. Prior knowledge and new task involve different persons, they are also recorded with a different camera but the scene remains the same. Specifically we considered Seq $3a$, Seq $1b$ and Seq $1c$. Figure 3.5 (c) shows the results: here *Transfer* is still significantly better than *No Transfer* but the gain in terms of recognition performance is small.

Case 4: different person, different camera, different scene. Finally we consider a prior knowledge setting where the person, the camera used and the



(a) Case 1 *(b)* Case 2

(c) Case 3 *(d)* Case 4

Figure 3.5*: Average recognition rate results on ten runs evaluated varying the number of samples per class in the training set. The significance of the comparison between Transfer and No Transfer is evaluated through the sign test and a square marker is reported on the graph if* $p < 0.05$*. Passing from case 1 to case 4 the prior knowledge is less and less relevant, consequently the advantage of Transfer w.r.t. No Transfer decreases.*

background scene are different with respect to the one used in the training and test videos. We used Seq $4a$, Seq $1b$ and Seq $1c$ and the results are reported in Figure 3.5 (d). Here the transfer learning system automatically realizes that the information coming from prior knowledge is not useful for the new task and *Transfer* performs as *No Transfer*.

Comparing all the four graphs in Figure 3.5, the progressively lower relevance of prior knowledge with respect to the new target task can be read in the decreasing recognition rate result for the *Initial Model*. Globally, the classifiers

obtained with *Transfer* learning perform better or at least equally to *No transfer*. Therefore we use the transfer learning to fix the activity class labels that are delivered to update the activity trackers.

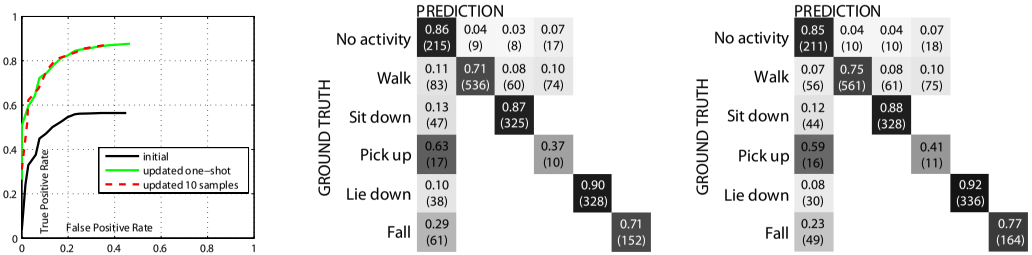
3.4.3 Activity Tracking

Given an updated set of activity trackers, we evaluate how the activity recognition performance increases with respect to the initial processing. The predicted activities are compared to the ground truth. We use Seq $*b$ for evaluation since it was not used previously for testing the classification. Activities are predicted for three cases: (*i*) the initial tracker set, (*ii*) the tracker set after the transfer learning update with one-shot learning and (*iii*) after the update with 10 manually labeled frames.

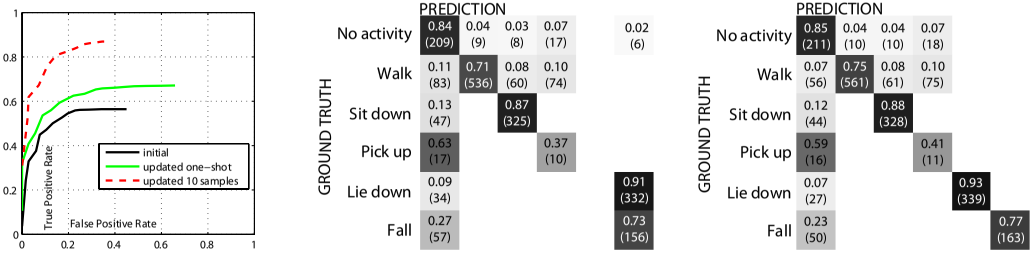
In Figure 3.6 we provide detailed insights for the activity update. The cases 1 (same scene, same person, same camera) and 4 (different scene, different person, different camera) are depicted. In Figure 3.6, ROC curves are shown for the initial and updated (one-shot and 10 samples) tracker sets. To this end, the threshold that determines the active trackers, is gradually increased. This results in different numbers of true-positives and false-positives. For the confusion matrices in Figure 3.6 and all further experiments, the threshold is kept fix.

One-shot labeling already improves the activity tracking performance considerably with respect to the initial tracker set. If the labels provided by the one-shot learning are correct as in case 1, the benefit of labeling 10 frames is marginal. If it turns out that one manually labeled sample is not sufficient for a good classification accuracy, as in the most difficult case 4, manual annotation of 10 frames improves the final performance. In the confusion matrices, the predicted activities are reported *vs.* the ground truth in terms of number of frames and underlie this finding. Cases 2 and 3 are very similar to case 1, *i.e.* the transfer learning with one manually labeled sample is sufficient.

In Table 3.2, we report the evaluation of the activity recognition in terms of overall true-positive-rate and false-positive-rate for different cases of prior knowledge and target tasks. The first four columns report results obtained on the same sequences used for the experiments in Figure 3.5, the last two columns contain the results for other test sequences. In all cases, the augmentation of the tracker set with new trackers learned from the transferred labels helps. In



(a) Case 1: ROC, confidence matrices for learning with 1 (left) and 10 (right) annotated samples



(b) Case 4: ROC, confidence matrices for learning with 1 (left) and 10 (right) annotated samples

***Figure 3.6**: Activity tracking results. ROC curves and confusion matrices for case* 1 *(top row, corresponding to Figure* 3.5*(a)) and case* 4 *(bottom row, corresponding to Figure* 3.5*(d)). In the first row, the performances for one-shot learning and learning with* 10 *samples match, whereas in the more difficult case in the bottom row, more annotations improve the performance.*

Tracker test sequence		$1b$				$3b$	$2b$
Tracker update sequence		$1c$				$3c$	$2c$
Transfer prior sequence		$1a$	$2a$	$3a$	$4a$	$3a$	$1a$
Corresponds to case		1	2	3	4	1	2
Initial processing	TPR	0.50				0.45	0.44
	FPR	0.13				0.17	0.06
Updated (1-shot)	TPR	0.78	0.78	0.78	0.59	0.72	0.62
	FPR	0.14	0.14	0.14	0.16	0.17	0.05
Updated (10 samples)	TPR	0.81	0.82	0.81	0.81	0.73	0.66
	FPR	0.15	0.17	0.15	0.15	0.22	0.13

***Table 3.2**: Results for different sequences, the predicted activity is compared to the ground truth. Different cases are reported in terms of true positive rate (TPR) and false positive rate (FPR). The updated activity set outperforms the initial one. In most situations, the results obtained with 10 labeled samples are only marginally better than using one-shot learning.*

five of the six evaluated cases however, the annotation of ten frames *vs.* one frame only improves the performance marginally. We underline that the number of labeled training samples needed is in any case two or at least one order of magnitude smaller than what originally requested to train the activity trackers in Section 2.3.

3.5 Conclusions

Starting from the output of the tracker-tree that detects known activities unusual events in surveillance videos, we presented here a strategy to learn these new events. These events can be new activity concepts or also abnormal, but relevant events that shall be recognized in the future. We only need a very small number of training samples since we exploit prior knowledge of activities that were known already. The intermediate transfer-learning process serves as artificial expert and permits the accurate labeling of multiple video frames. We are then able to integrate the new activity concepts in the tracker-tree besides the existing ones and hence improve the activity detection performance of the tracker tree during runtime.

4

Unsupervised Behavior Analysis in Two Hierarchies

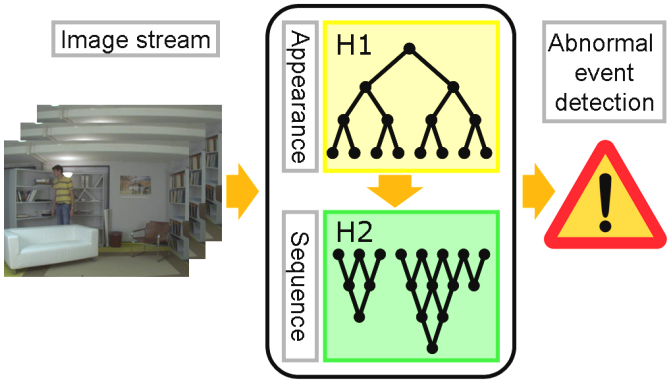
4.1 Introduction

In the previous chapter, we presented an approach to incrementally learn new actions during runtime, that requires manual labeling of few activity samples. Here we go one step further and propose to model the observed behavior without any human assistance. In the context of in-house monitoring, one faces the challenges that every person behaves differently and the camera setup and the room layout changes for every installed system. Therefore, a model learned off-line in lab-settings is likely not to represent the observations well, and abnormal event detection might fail in practice. Due to these reasons, we are interested in modeling the observed behavior of the actually concerned person as accurately as possible. In this chapter, we propose a technique inspired by biological findings, that learns human behavior in a completely unsupervised manner.

In a biological perspective, it is of utmost importance for the survival of all animals to correctly recognize motion. In particular, human locomotion consists of motion patterns that involve different movements of all limbs and are therefore widely used to study visual motion perception on a psychophysical level but also for modeling with algorithms. Recent computational techniques to model human motion therefore incorporate biologically motivated strategies (*e.g.*, [Giese and Poggio 2003, Jhuang *et al.* 2007, Escobar *et al.* 2009]. In a different study, Lange and Lappe [Lange and Lappe 2006] propose a neurally plausible model that explains the visual perception of biological motion. According to their findings, snapshot responses of the first stage (form) are tem-

(a) Normal action

(b) Abnormal event



(c) Human behavior analysis

***Figure 4.1**: 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.*

porally integrated in the second stage (motion). An analogue concept can be relied on for computational action modeling as for example done in [Schindler and Van Gool 2008].

Our model consists of two hierarchical representations arranged in a cascade, as illustrated in Figure 4.1(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 from 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 an-

alyze 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 (Section 4.2). Our approach is additionally motivated by the recent work of [Lin *et al.* 2009] which also relies on a hierarchical representation, but targets action recognition in a supervised setting.

At runtime, the learned structure is used as a model of normality to which unseen data is compared (Section 4.3). The person is tracked and statistical outliers with respect to appearance and action are detected robustly and efficiently at different hierarchical levels. We additionally show how to update this model in order to incorporate newly observed normal instances. Experiments in Section 4.4 target the surveillance of humans in indoor environments and show abnormal event detection capacities of our approach.

To underline the biological relevance of our approach, we compare the performance of our computational model to responses of monkeys for the task of discrimination of locomotion direction (Section 4.5). Trained and evaluated on the same input data, we show parallels and shortcomings between monkeys behavior and the two-stage computational model.

The elaborations in this chapter are based on [Nater *et al.* 2010a] and [Nater *et al.* 2010b].

4.2 Human Behavior Modelling in Hierarchies

In the following, we present our approach to model human behavior in two hierarchies, dedicated to encode the appearance and the motion of the person, respectively.

4.2.1 Appearance Hierarchy ($H1$)

We start from an image stream

$$\mathcal{S} = \langle \boldsymbol{x}_1, \ldots, \boldsymbol{x}_T \rangle, \quad \boldsymbol{x}_t \in \mathcal{X} \tag{4.1}$$

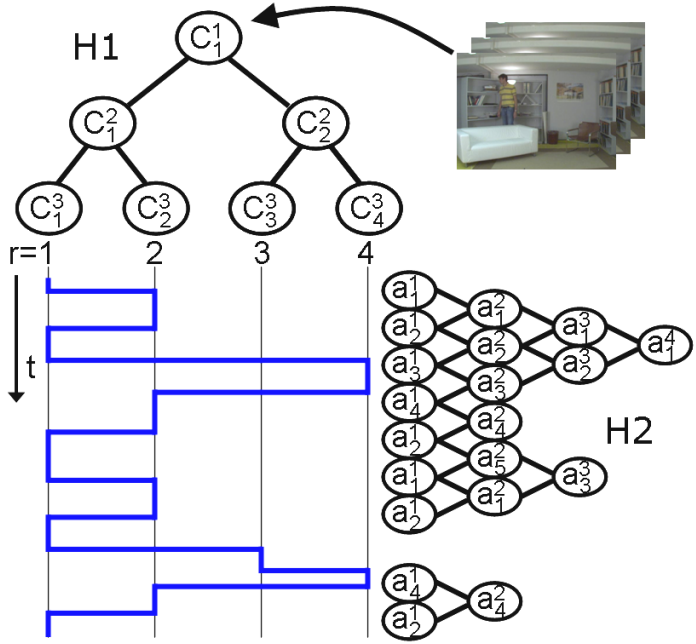


***Figure 4.2**: Illustration of the unsupervised learning approach, composed of two hierarchies. In $H1$, a sequence of images is mapped by clustering to a number of discrete symbols, in $H2$ the sequence of these symbols is analyzed.*

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 [Jain *et al.* 1999], applied hierarchically to the training data in a top-down procedure with a distance measure $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ defined in $\mathcal{X}$. The root node cluster $\mathcal{C}^{(1)}$ describes all $\boldsymbol{x}_t \in \mathcal{S}$. 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 $l+1$:

$$\mathcal{C}^{(1)} = \mathcal{X}, \quad \mathcal{C}_i^{(l)} = \bigcup_{k'=1}^{k} \mathcal{C}_{k'}^{(l-1)}. \tag{4.2}$$

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 Figure 4.2 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 $\boldsymbol{x}_t$ 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 $\boldsymbol{x}_t$ 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.*

$$\mathcal{S} \mapsto \mathcal{R} = \langle r_1, \ldots, r_T \rangle, \quad r_t \in \mathbf{N} \cup \{\sharp\}. \tag{4.3}$$

In order to obtain compact clusters and to cope with noisy data, we remove statistical outliers at every clustering step with the formulation of Section 4.3.1. The symbol $r = \sharp$ is assigned to an $\boldsymbol{x}_t$ 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 $\boldsymbol{c}_i$ and the distribution $D_i^{(l)}$ of distances $d_i = d(\boldsymbol{c}_i, \boldsymbol{x})$ of all the samples $\boldsymbol{x}$ assigned to this cluster.

4.2.2 Illustration

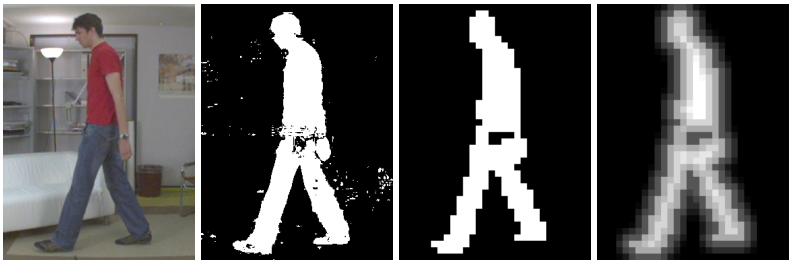
We demonstrate the mapping of input images to clusters in the tree structure. An indoor training sequence[1] of about $7,100$ images was recorded at 15 frames per second in VGA resolution. It contains diverse 'every-day' actions such as walking, walking behind occluding objects, sitting on different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

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

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

[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.

is applied. Maximum and minimum pixel values are bounded and an offset is added to obtain non-negative values (*c.f.* Figure 4.3). Finally, the rows are concatenated in a vector that defines the fixed length image features $\boldsymbol{x}$ ($N = 1600$), describing the appearance of one person in the scene.



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

***Figure 4.3**: Feature extraction: (a) original, (b) segmented, (c) postprocessed 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 [Belongie *et al.* 2002]. Two samples $\boldsymbol{x}_u$ and $\boldsymbol{x}_v$ with elements $x_u(n)$ and $x_v(n)$, $n = 1 \ldots N$ are at a distance

$$d(\boldsymbol{x}_u, \boldsymbol{x}_v) = \frac{1}{2} \sum_{n=1}^{N} \frac{[x_u(n) - x_v(n)]^2}{x_u(n) + x_v(n)}. \tag{4.4}$$

This said, the silhouette features are extracted and clustered ($k = 2$) in order to build $H1$. The outcome is visualized in Figure 4.4, 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.

4.2.3 Action Hierarchy ($H2$)

As depicted in Figure 4.2, we start from the sequence of symbols $\mathcal{R}$ defined in Equation (4.3). The goal is to exploit the information in this sequence and extract frequent patterns which we refer to as micro-actions. Their variable length

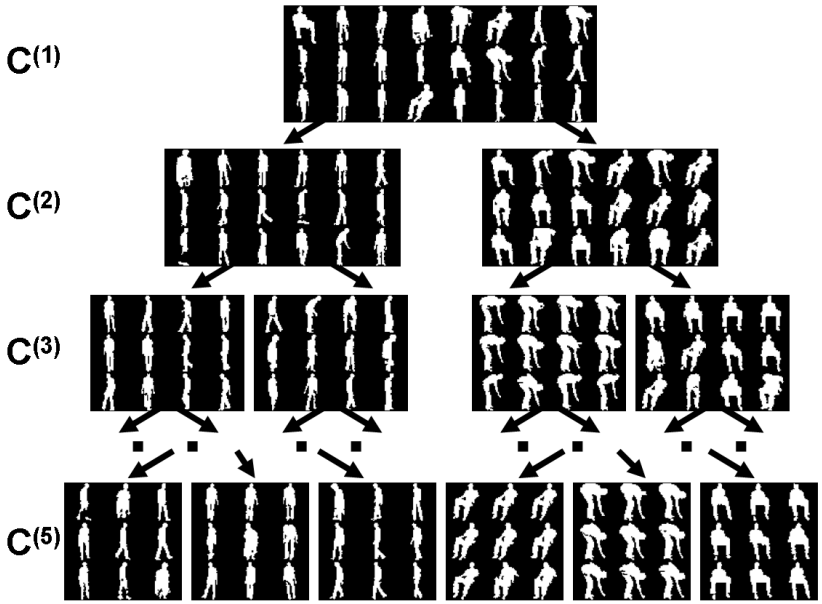


***Figure 4.4**: Visualization of the proposed binary tree for the hierarchical appearance representation ($H1$). For each of the displayed clusters at different layers $C_i^{(l)}$, randomly chosen silhouettes are displayed.*

naturally defines a hierarchy, since longer actions automatically represent more information. Our approach is inspired by the work of Fidler *et al.* [Fidler *et al.* 2006, Fidler and Leonardis 2007], 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 and noise is omitted. 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.

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:

$$\mathcal{A}^{(1)} = \{a_i^{(1)} := r_t \rightarrow r_{t+1} \mid r_t \neq r_{t+1}, P(a_i^{(1)}) > \theta_{act}\}, \qquad (4.5)$$

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.*

$$\mathcal{A}^{(\lambda)} = \{a_i^{(\lambda)} := a_p^{(\lambda-1)} \rightarrow a_q^{(\lambda-1)} \mid P(a_i^{(\lambda)}) > \theta_{act}\}. \tag{4.6}$$

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:

$$\mathcal{S} \mapsto \mathcal{R} \mapsto \langle a_1^{(\lambda)}, \ldots, a_t^{(\lambda)} \rangle, \quad a_i^{(\lambda)} \in \mathcal{A}^{(\lambda)}. \tag{4.7}$$

Note that in this formulation, micro-actions can overlap, which is in line with the observation that often no clear-cut boundaries of actions can be defined [Natarajan and Nevatia 2008, Satkin and Hebert 2010].

4.2.4 Illustration

Action recognition We employ a publicly available dataset [Lin *et al.* 2009] that is previously used for action recognition. To illustrate the extraction of micro-actions, we 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 Figure 4.5. Binary silhouettes are provided in the dataset, we quantize them in the same way as in Section 4.2.2. In a first step, the appearance hierarchy of Section 4.2.1 is grown and the plotted sequence in

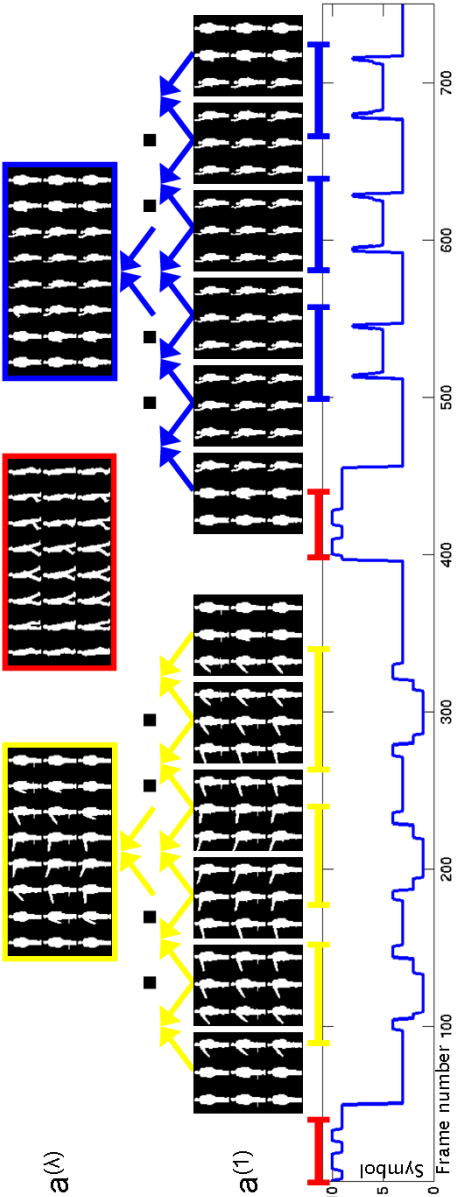


***Figure 4.5**: Illustration of the micro-action hierarchy (H2) for the action recognition test dataset of [Lin et al. 2009]: Micro-actions are extracted from symbol transitions and can be combined gradually into higher level actions.*

the bottom of Figure 4.5 depicts the obtained symbols. Next, repeated patterns in this sequence are extracted first on the basic level $a^{(1)}$ (*i.e.* transitions, Equation (4.6)), then growing in length on higher levels (Equation (4.7)). The finally meaningful micro-actions are presented in the upper part of Figure 4.5 and correspond to the actions which are to be recognized, but discovered in an unsupervised manner.

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 Figure 4.6. 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.

4.3 Runtime Processing

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.

4.3.1 Data-dependent Inlier

Given a query image with extracted features $\boldsymbol{x}$, we want to determine its cluster membership $\mathcal{C}_i$ based on the distance $d(\boldsymbol{x}, \boldsymbol{c}_i)$. 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 [Knorr and Ng 1998], comparing $d(\boldsymbol{x}, \boldsymbol{c}_i)$ to the distance distribution D_i of the cluster $\mathcal{C}_i$. For each cluster $\mathcal{C}_i$ with center $\boldsymbol{c}_i$, this distribution D_i of the distances $d_i = d(\boldsymbol{c}_i, \boldsymbol{x}_t)$ for all training samples $\boldsymbol{x}_t$ that were assigned to this

(a) 'walk'

(b) 'sit'

(c) 'pick up'

***Figure 4.6**: 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.*

cluster was estimated during training. The probability that the query point $\boldsymbol{x}$ is an inlier to $\mathcal{C}_i$ is

$$p_{inlier}\left(d(\boldsymbol{x}, \boldsymbol{c}_i)\right) = 1 - \int_{\xi=0}^{d(\boldsymbol{x},\boldsymbol{c}_i)} D_i(\xi) d\xi. \tag{4.8}$$

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

$$p_{inlier}\left(d(\boldsymbol{x}, \boldsymbol{c}_i)\right) \geq \theta_{inlier}. \tag{4.9}$$

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

4.3.2 Target Tracking

In every frame we want to determine the location and scale of the bounding box, *i.e.* find $\boldsymbol{x}_t$ that best matches the trained model. This is important for a correct and robust 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 $\boldsymbol{x}'_t$ extracted from a hypothesized location and scale is evaluated by using Equation (4.9) and is propagated as far as possible in $H1$, from the root towards the leaves. With this inlier formulation, an observed image representation $\boldsymbol{x}'_t$ 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 $\boldsymbol{x}_t$, 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 $\boldsymbol{x}_t$.

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, it is simultaneously followed by a generic foreground object tracker, which specifies the bounding box in this case. To this end, we use the same mode estimating tracker [Bradski 1998] as in the root node of the tracker-tree in Chapter 2. In our current implementation, this tracker is also used to establish a prior for the exhaustive search, which additionally speeds up the tracking procedure.

4.3.3 Abnormal Appearance

An abnormal (or novel) appearance is identified in $H1$ on hierarchical level l if the tracking result $\boldsymbol{x}_t$ 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 $\boldsymbol{x}_t$ in this case, the symbol $r_t = \sharp$ is attributed, characterizing an unknown (not matching) state. Of course, if $\boldsymbol{x}_t$ is outlier at the root node already, it is also abnormal. This outlier detection paradigm is in line with state of the art novelty detection in a disjunctive hierarchy [Weinshall *et al.* 2012]. This additionally motivates the use of a hierarchical structure for data modeling.

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.

This said, novelty detection in $H1$ is based exclusively on the appearance per frame. The identification of abnormal actions is achieved in $H2$.

4.3.4 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 Equation (4.5) and Equation (4.6) 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 $\boldsymbol{x}_t$ is mapped to the unknown state $r_t = \sharp$, no micro-action can be established and the sequence analysis breaks down temporarily. This is due to the prerequisite that usual actions require a sequence of usual appearances.

4.3.5 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. For example, lying on the couch could be observed often, whereas lying on the floor not. The same idea applies to actions performed at a certain time of day, for example, a person observed walking through a living room at 4 a.m. should not necessarily be considered normal.

This family of unusual events could be accounted for by learning statistics on the spatial and temporal extent of the previously established symbols and micro-actions. However, the incorporation of such techniques into our method is not the focus of this thesis.

4.3.6 Model Update

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, for example 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.

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, for instance 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} (Equation (4.9)) to a high value already for clustering. Finally, new leaf node clusters are established and new symbols are defined.

Micro-action update Established micro-actions by definition have a sufficient frequency of occurrence (Equation (4.6)). We propose to estimate these probabilities incrementally, by updating them with new observations during runtime, 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 (Equation (4.5)) 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.

4.4 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 which are meaningful 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.

4.4.1 Behavior Analysis

The training video recorded in the indoor environment was already introduced in Section 4.2.2. The test footage of about $1,000$ images was recorded in the same setting as the training sequence, 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 Figure 4.4, extraction of frequent micro-actions like the ones in Figure 4.6), 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 Figure 4.7.

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 Equation (4.9). 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 [Bradski 1998]), nuances of orange are used for intermediate layers and green encodes an appearance 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 length of the bar is adjusted 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 in Figure 4.8 indicate three temporal characteristics:

- 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.
- The matching cluster identity (symbol r_t) changes over time ($0 = \sharp$) which allows for the recognition of micro-actions.

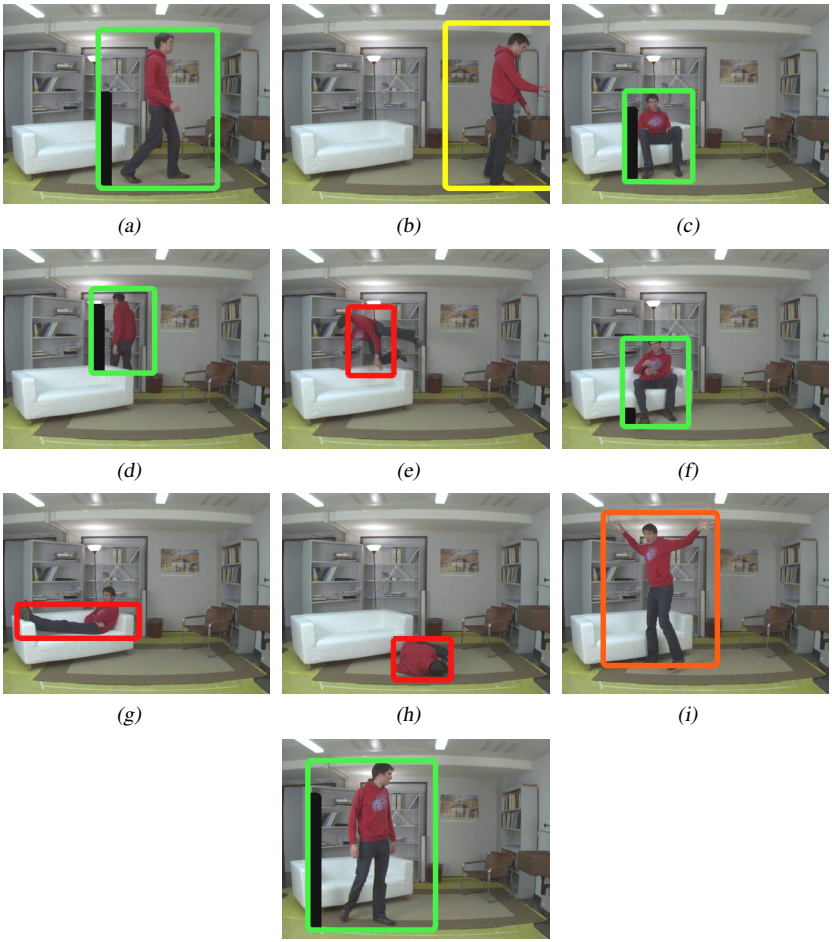


Figure 4.7: *Our method tracks the person, analyzes the appearance in $H1$ and interprets the micro-action in $H2$. Here we present various normal and abnormal instances of the test sequence. The color of the bounding box encodes the layer in $H1$, on which the observation is normal, the length of the black bar on the left side of the bounding box indicates the micro-action level in $H2$.*

- The micro-actions are matched hierarchically and the maximal length is plotted.

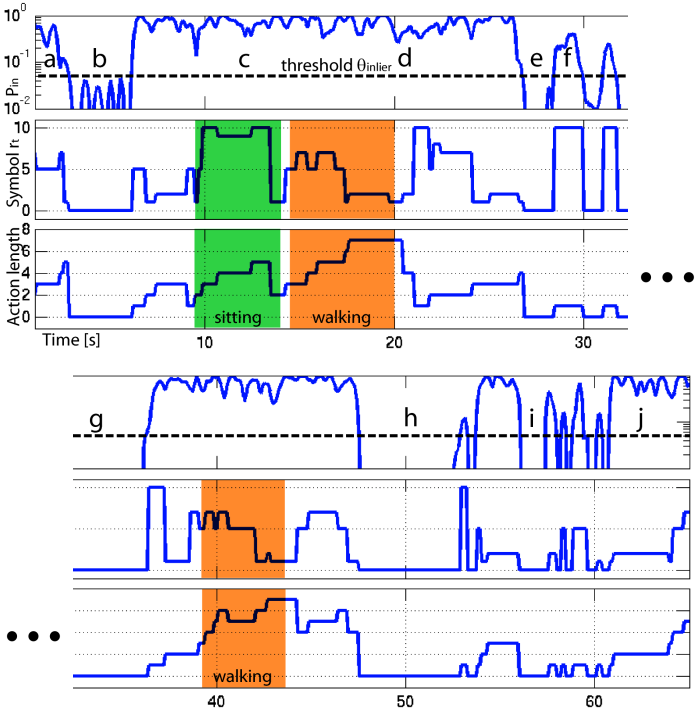


Figure 4.8: *Three representative values are plotted over time, the inlier probability at the leaf node level of $H1$, the matched symbol r_t and the micro-action length $a^{(\lambda)}$. Two actions are highlighted (see text for details).*

Two patterns ('walking' and 'sitting') are highlighted in color, which in fact correspond to the same micro-actions as shown in Figure 4.6(a) and Figure 4.6(b). The frames of Figure 4.7 are localized in time with the letters in the top plot of Figure 4.8.

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. This also demonstrates the use of the hierarchy: the leaning person in (b) is detected abnormal close to the leaves, while the waiving in (i) is more severe but still an upright pose. Severe

abnormalities are detected at the root node. 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 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.

4.4.2 Model Update

(a)

(b)

(c)

(d)

Figure 4.9*: Illustration of the update procedure: (a) some feature cropped image regions and their corresponding feature representations used for the update of $H1$, (b) normal appearances stay normal after the update, (c) lying turns normal after the update and (d) real outliers are still detected.*

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 some randomly chosen outlier samples are displayed in Figure 4.9(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 (Figure 4.9(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 Figure 4.9(c).

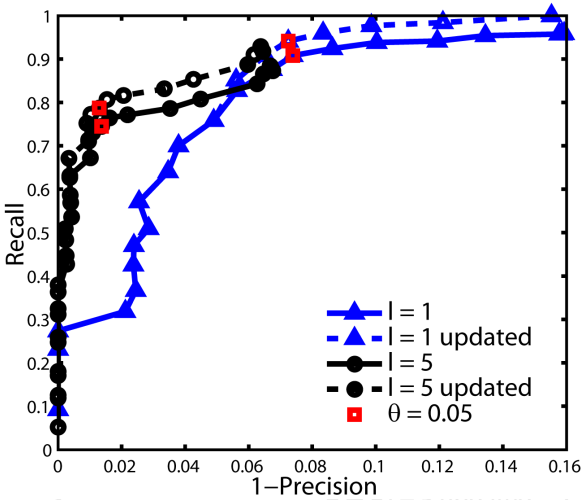


***Figure 4.10**: Recall-precision curves for the video sequence of Figure 4.7 verifies the applicability of our technique.*

To quantize the experimental results, we manually annotated abnormal events per frame for the sequence of Figure 4.8. A RPC plot, depicted in Figure 4.10, measures the performance by sweeping parameter θ_{inlier} (Equation (4.9)) for different model configurations. The benefit of a hierarchical model is apparent when the two model depths are compared. Indeed, moving down in $H1$ (from layer 1 to layer 5) increases the precision dramatically. This is essential for our task. Also the effect of the model update that includes the lying poses is observed. More precisely, this means that at a precision of 98%, the recall

increases from 32% (root node level) to 78% (leaf node level), respectively to 81% after the update.

Although these measures demonstrate the abnormal event detection capacity of our method, they have a limited significance. In fact, they only capture the abnormality detection performances at a single layer in the hierarchy. The real benefit of the hierarchical model is the detection of outliers at the different layers. Hence, a fall (detected at the root, as depicted in Figure 4.7(h)) is considered more a severe abnormal event compared to the slightly inclined hand-washing in Figure 4.7(b) that is an outlier only at the leaves. Unfortunately it remains unclear how to integrate these subjective semantic concepts into a quantitative measure.

4.5 Behavioral Relevance

In this section, we demonstrate the behavioral relevance of the human behavior model, described in this chapter. This is motivated by the fact, that the correct recognition of behavior is of utmost importance for the survival of animals. Hence, we investigate how our algorithms compare to the capacities of macaque monkeys for analyzing simple walking patterns. In particular, we investigate the discrimination of human locomotion direction.

While discrimination between right- and leftward walking is possible based on shape cues only, forward and backward walking requires motion for a successful distinction [Lange and Lappe 2006]. A recent behavioral study in macaque monkeys investigated the perception of walking direction and how well these animals generalized from a trained categorization of walking to other walking speeds and running [Vangeneugden *et al.* 2010]. The question now arises how our previously developed technique relates to these findings. This is particularly interesting, as we also rely on the separated encoding of typical appearance and motion patterns. We compare the behavior of our algorithm to the monkeys behavior with the same input stimuli for the task of human locomotion coding at different walking and running speeds. We first introduce the test setup and the tasks, then show how our computational model is used and finally present the experiments.

This section was the result of an inspiring collaboration with Joris Vangeneugden and Rufin Vogels at K.U. Leuven, Belgium. It was published in [Nater *et al.* 2010b].

4.5.1 Subjects and Apparatus

Three rhesus monkeys (*Macaca mulatta*) M1, M2 and M3 served as subjects in this study. The heads of the monkeys were kept immobilized during the sessions (approx. 3h/day) in order to capture the position of one eye via an infrared tracking device (EyeLink II, SR Research, sampling rate 1000 Hz). The recording setup is displayed in Figure 4.11. Eye positions were sampled to assure that the subjects fixated the stimuli. In order to obtain a juice reward (operant conditioning), successful fixation, within a predefined window measuring $1.3° - 1.7°$, and a correct saccade towards one of the response targets were required (*c.f.* Figure 4.12). More specifically, each trial started with the presentation of a small red square at the center of the screen (size $= 0.12 \times 0.12$). The subjects had to fixate this square for 500 ms, followed by the presentation of the stimulus (duration $= 1086$ ms; 65 frames at a 60 Hz frame rate). Before making a direct eye movement saccade to one of the two response targets, the monkeys had to fixate the small red square for another 100 ms. During the complete trial duration, monkeys had to maintain their eye position within the predefined window. Failure to do so resulted in a trial abort. Response targets were located at $8.4°$ eccentricity, either on the right, left of upper part of the screen. The stimuli, described below, measured approximately $6°$ by $2.8°$ degrees (height/ width at the maximal lateral extension respectively).

All animal care, experimental and surgical protocols complied with Belgian and European guidelines and were approved by the K.U. Leuven Ethical Committee for animal experiments.

4.5.2 Stimuli

Stimuli were generated by motion-capturing a male human adult of average physical constitution walking at 2.5, 4.2 or 6 or running at 8, 10 or 12 km/h. Specifications of the procedure can be found in [Vangeneugden *et al.* 2010]. Eenriched stimulus versions were rendered by connecting the joints, *i.e.* coordinates, of an otherwise invisible agent by geometrical cylinder-like primitives, hence forward labeled as humanoids. Importantly, all stimuli were displayed resembling treadmill locomotion, devoid of any extrinsic/translatory motion component (*c.f.* Figure 4.13).

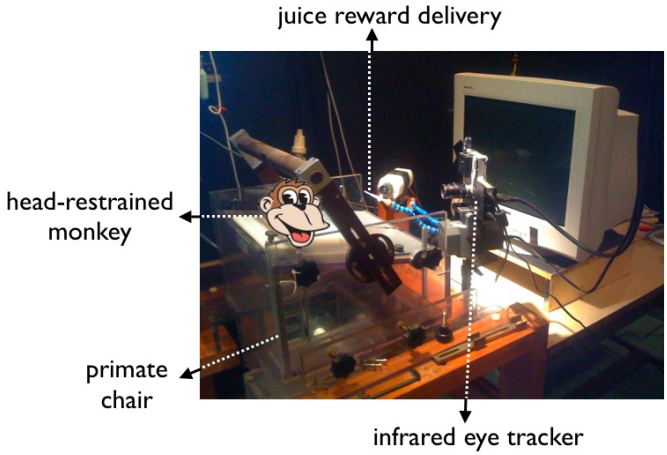
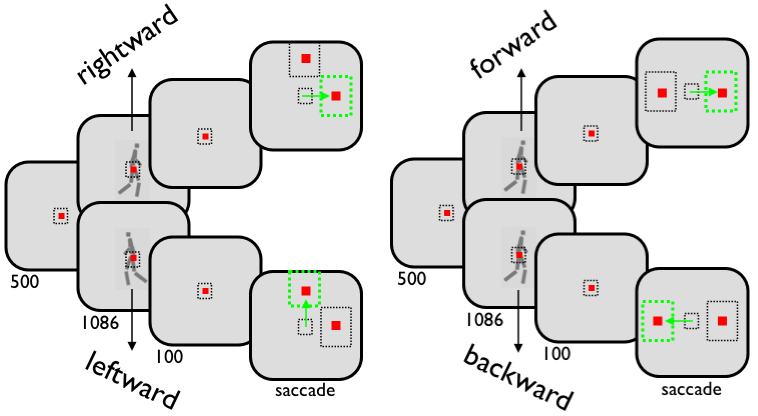


***Figure 4.11**: Recording setup for the behavioral study. (Figure credits: Joris Vangeneugden)*

4.5.3 Tasks and Training

The three monkeys were extensively trained in discriminating between different locomotion categories. In a first task, they were instructed to discriminate between different facing directions (LR/RL task shown in Figure 4.14(a)) when observing the stimulus (video) that shows a person that is either walking towards the right (LR_fwd) or towards the left (RL_fwd). The second task is designed to distinguish forward from backward locomotion (FWD/BWD task shown in Figure 4.14(b)). In that case, the stimulus shows a person walking towards the right, but either forward (LR_fwd) or backward (LR_bwd). The LR_bwd condition was generated by playing the LR_fwd video in reverse. The start frame of the movie stimuli was randomized across trials to avoid that the animals responded to a particular pose occurring at a particular time in the movie. Training was done only at 4.2 km/h walking speed.

Substantial training was needed for our monkeys to learn FWD/BWD discriminations, while LR/RL discrimination was made more easily (*c.f.* [Vangeneugden *et al.* 2010]). *E.g.*, the number of trials required to reach 75% correct in a session for the LR/RL task was 1323 trials, while the same monkey required $37,238$ trials to achieve a similar performance level in the FWD/BWD task.



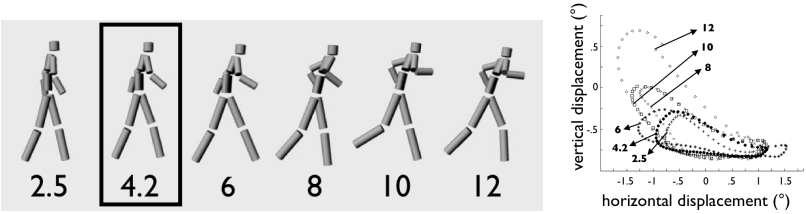
(a) LR/RL task *(b)* FWD/BWD task

***Figure 4.12**: Illustration of the recording procedure. The gray fields indicate what was presented to the monkeys on the screen, with the respective durations indicated below each screen-shot. The dotted rectangles around the blue fixation targets represent the windows in which the eye movements (saccades) had to land. In green are the correct targets the monkeys had to saccade to in order to obtain a juice reward. The timing in milliseconds is displayed beneath the gray screens.*

Nevertheless, all three monkeys reached behavioral proficiency at the end of the training sessions.

4.5.4 Generalization Test

Trained at one speed only, the monkeys were tested for generalization to other speeds in the two described tasks. This was realized by interleaving trials of the trained speed with trials from the other speeds in a 90:10 ratio. Moreover, in order to avoid associative learning on these new stimuli, we always rewarded the monkeys on these other speed stimuli (still correct responses on the trained speed were required to obtain a juice reward).



(a) stimuli for the 6 speeds *(b)* ankle trajectories

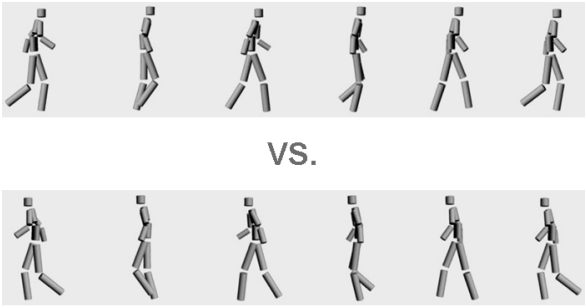
***Figure 4.13**: Stimuli presented in the behavioral and the computational experiments. In (a), snapshots of the 6 walking speeds are depicted and the training speed is framed. (b) shows the ankle trajectories for the same speeds. With increasing speed, step size increases as well as vertical displacements grow.*

4.5.5 Computational Model

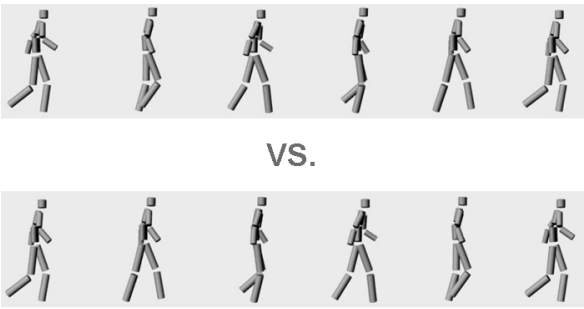
Training We use our technique described previously in this chapter to encode the human behavior in appearance (*H1*) and motion (*H2*) hierarchies. To train the model, we use the 4.2 km/h stimuli as depicted in Figure 4.14, transformed to the silhouette-based representation. Since our model is designed to cope with larger amount of data, 6 repetitions of the same stimuli were used for training. The learned appearance hierarchy (*H1*) consists of 4 layers resulting in 8 leaf node clusters. Separate models were trained for LR_fwd, RL_fwd and LR_bwd.

Generalization test During testing, we use the LR_fwd and the RL_fwd models for the LR/RL task, whereas in the FWD/BWD task we apply the LR_fwd and the LR_bwd models. New stimuli at different walking and running speeds are presented to the model in order to test the generalization capacity.

Each model applied to the test data delivers two output values per frame, that characterize how well each test frame matches *H1* and *H2* (assuming *H1* has validated the stimulus), respectively. The value for *H1* captures the appearance only by measuring the similarity to one of the leaf node cluster centers. Additionally, *H2* requires the correct motion and searches for the a corresponding micro-action with maximal length. To finally achieve the output score (appearance score from *H1*, sequence score from *H2*) we combine two models with contrary training. They are evaluated at each frame and a likelihood ratio is



(a) LR/RL task: LR_fwd *vs.* RL_fwd



(b) FWD/BWD task: LR_fwd *vs.* LR_bwd

Figure 4.14*: Selected frames of the sequence stimuli presented for the LR/RL and the FWD/BWD task at* 4.2 *km/h walking speed*

calculated and averaged across the whole stimulus. This results in one output value per evaluated speed for *H1* and *H2*. If for example a stimulus with walking from left to right is described well in LR_fwd, but not in RL_fwd, its score is high. On the other hand, if both models perform equally well, no clear decision can be drawn and the score is in proximity of 1 (chance level).

4.5.6 Experiment 1: LR/RL Task

The results are depicted in Figure 4.15, the monkeys responses are shown in panel (a), the appearance score (*H1*) in panel (b) and the sequence score (*H2*) in (c). Bold lines indicate the average results, dotted ones display individual

performances for monkeys or different stimuli. Black boxes at 4.2 km/h point out the training speed. Chance level is marked with the dashed horizontal line.

In the behavioral study (Figure 4.15(a)), categorization generalizes relatively well across the different walking and running speeds (binomial tests; $p < 0.05$ for 14 out of 15 generalization points). This suggests that the discrimination is based on spatial or motion cues that are common to the different speeds.

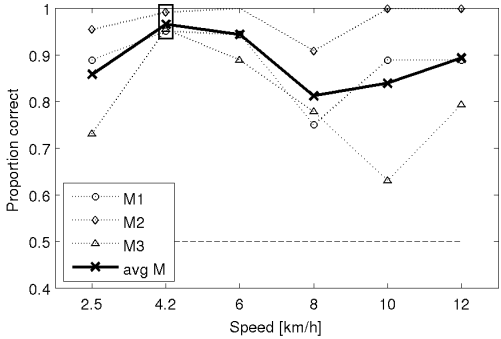
For the computational part, the results for *H2* (Figure 4.15(c)) indicate a similar interpretation for slower walking speeds (2.5-6 km/h). In a more detailed analysis, we observe that for these speeds, the task is already solvable in *H1*. Apparently, the appearances are distinctive enough. For the running stimuli on the other hand, silhouettes are different, thus the performance in $H1$ drops, which also influences *H2*. Further, in *H1* the trained walking speed clearly outperforms all the other speeds. The data-driven machine learning approach does not generalize as well as monkeys are able to.

4.5.7 Experiment 2: FWD/BWD Task

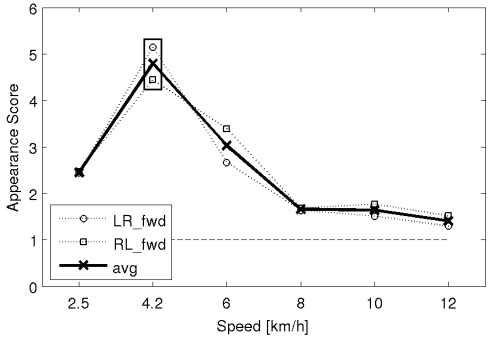
The results for the FWD/BWD task are visualized in Figure 4.16 in the same manner as for the previous task. The behavioral data from the speed-generalization tests shows that the categorization is specific to walking: in each monkey, generalization is significant (binomial test: $p < 0.05$) for the walking, but not the running patterns. In fact, in each monkey there is an abrupt drop of the performance when the locomotion changes from walking to running.

Computational findings show that the evaluation of the appearance exclusively is not sufficient for solving this task (Figure 4.16(b)). This is not surprising, since the appearances are the same for both stimuli. However, if their ordering is considered (Figure 4.16(c)), the task is solvable for walking speeds, and the scores resemble a lot monkeys responses. At higher speeds, due to wrong appearance classification in $H1$, the sequence is not reliable in $H2$ anymore.

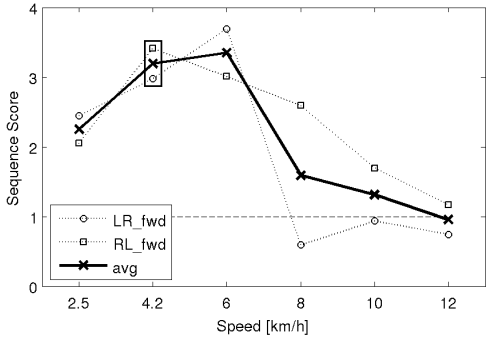
The lack of significant transfer from the trained walking to running suggests that the animals learned a particular motion trajectory "template". Indeed, examination of the ankle trajectories (*cf.* Figure 4.13(b)) reveals a relatively high similarity between those trajectories for the three walking speeds, which are in turn rather distinct from those of the three running patterns. This might also be a reason for the performance drop of the computational model.



(a) Monkeys responses

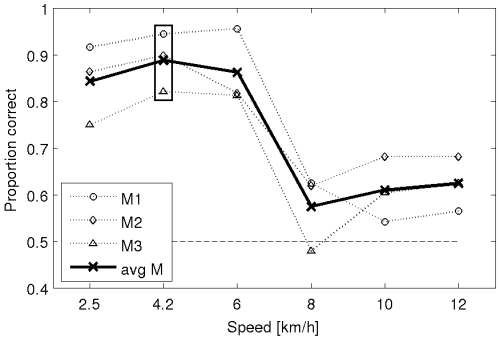


(b) H1 output

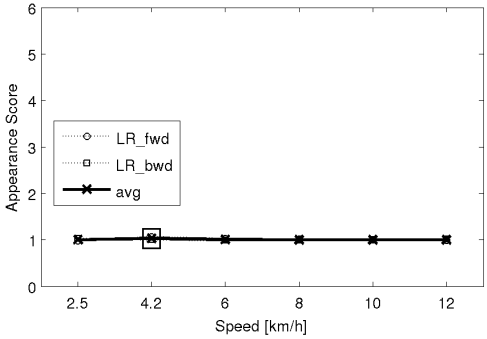


(c) H2 output

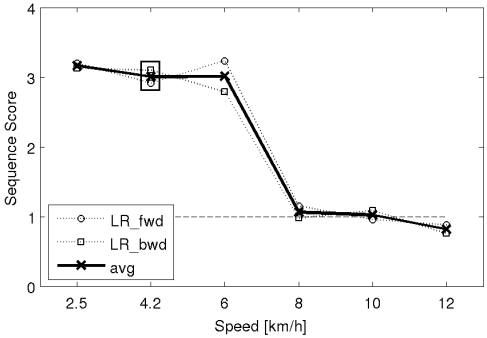
Figure 4.15*: LR/RL task: Monkeys performance and model scores for* 6 *tested speeds*



(a) Task



(b) H1 output



(c) H2 output

Figure 4.16*: FWD/BWD task: Monkeys performance and model scores for* 6 *tested speeds.*

4.5.8 Discussion

At the behavioral level in the monkeys we noticed a clear qualitative difference in generalization performances across tasks. Whereas the monkeys were quite apt at discriminating other speeds not seen before in the LR/RL task, a clear step-wise function was observed in the FWD/BWD task. In the LR/RL, when confronted with other walking speeds, *i.e.* 2.5 or 6 km/h, all three monkeys could correctly categorize these locomotions significantly higher than chance level. However this was not the case when confronted with locomotions at running speeds, a trend present in all three monkeys.

The broader generalization observed in RL/LR task compared to the FWD/BWD task shows that such motion cues are less specific. Alternatively, the monkeys might have used spatial features that are common to the walking and running humanoids that face in a particular direction. The fact that one could solve the view task quite simply by basing decisions on the presentation of just one frame could explain the observed (almost) perfect generalization. This is analogous to the first hierarchical analysis stage, which works on the per-frame appearances of actions. However, at this stage, the model shows a slightly different pattern, performing quite robustly for the trained locomotion, with clear drop-offs already for the neighboring speeds. This is clearly due to over-fitting of the model to the trained action. When implementing the second hierarchical stage of the model, which incorporates the evolution of the per-frame appearances over time, model's performance resembles the monkey's performances more closely, especially for the FWD/BWD task. Compared to the monkeys, the model not only picks up on the informative differences, *e.g.*, static information on the bending of the back, but takes into account all the pixel-level differences.

In summary, we see that monkeys have the capability to generalize well for simple tasks where snapshot information is sufficient. This might be due to prior knowledge based on different functional features, which is so far not included in the computational model at all.

4.6 Conclusions

We have presented an approach for the unsupervised analysis of human activity in surveillance 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.

In addition, we have presented a study that demonstrates the behavioral relevance as we were able to reproduce, at least to some extent, monkey's responses with our two-stage computational model. The algorithm however does not have the same generalization capacities which suggests that monkeys integrate the training in a (semantically) broader manner than the computer does.

5

Temporal Relations in Activity Analysis

5.1 Introduction

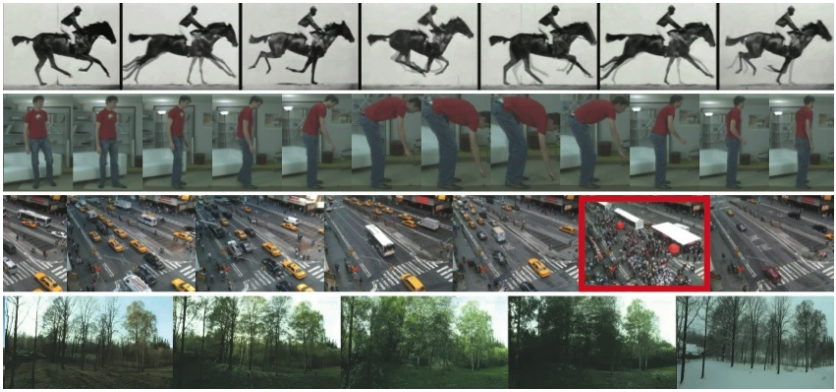


Figure 5.1*: In videos, each frame strongly correlates with its neighbors. Our approach exploits this fact and enables the segmentation of the video and the interpretation of unseen sequences.*

In Chapter 4 we have proposed a technique for unsupervised, bottom-up modeling of the behavior in a surveillance scene in order to detect abnormal situations. In the same category, techniques have been proposed in the literature, aiming at the interpretation of motion in public places (*e.g.* [Stauffer and Grimson 2000, Hospedales *et al.* 2009, Kuettel *et al.* 2010, Wang *et al.* 2009]), the

analysis of human actions as in [Niebles *et al.* 2008, Turaga *et al.* 2009] or the discovery of facial events [Zhou *et al.* 2010]. Unfortunately, such 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 or the structure of the model, as for the technique of Chapter 4) and/or, (iii) being too abstract for easy interpretation.

Here, we want to overcome the mentioned limitations of previous works by explicitly including the temporal characteristic of activities in videos during the model building and reasoning. Whereas the the technique of Chapter 4 separates appearance and motion into two distinct model components, we merge the two aspects in this chapter. In fact, we demonstrate how the inclusion of temporal information enables unsupervised activity discovery and precise activity modeling, such that abnormal events are robustly detected during runtime.

Observing the different sequences in Figure 5.1, one easily observes that increments between frames are quite small compared to the changes throughout the whole sequence. For instance, 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 hierarchical structure. Some observations might be salient, such as the big tent in a street festival (3^{rd} row in Figure 5.1). In this chapter, we explain our approach with respect to the modeling and interpretation of human activities, as shown in the sequence of the second row in Figure 5.1. In Chapter 6 we will show further applications in different surveillance scenarios.

The contributions made in this chapter 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 (Section 5.2). The automatically discovered activities are efficiently represented and continuously refined in a hierarchical manner (Section 5.3).
- Analysis and interpretation of unseen data is demonstrated as a result of the coarse to fine representation in the hierarchy. This enables the detection of abnormal events (Section 5.4).

Experimental results, presented in Section 5.5 show the usefulness of the proposed technique. Using human activity datasets, we demonstrate the segmen-

tation of the input video into meaningful activity snippets. The subsequent detection of abnormal situations based on the modeled activities outperforms the technique of Chapter 4.

The work presented in this chapter has been published in [Nater *et al.* 2011a].

5.2 Activities in Videos

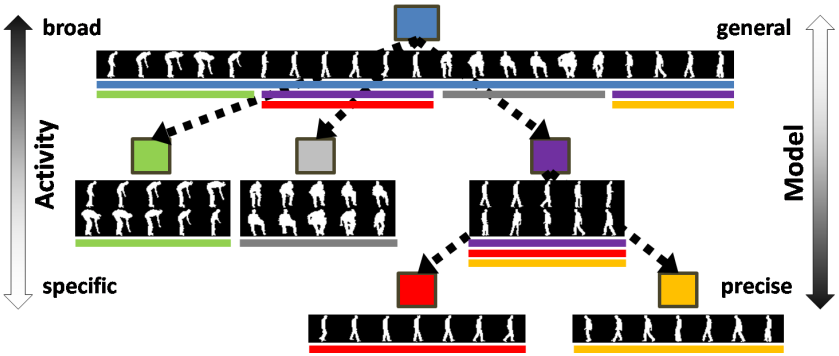


***Figure 5.2**: Overview of the proposed hierarchical model that splits and represents the data in a coarse to fine manner. As an example, we consider indoor actions. At the top node, the entire video stream is taken into account, while at lower levels, more specific concepts, like picking up, or walking leftwards are found.*

Due to the large variety of observations included in a surveillance video, 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. We call these subsequences *activities*. If they are consistent and have low complexity, they can be represented more easily and precisely. This principle is exploited by arranging the video data in a hierarchical manner as outlined in Figure 5.2. In a long data-stream, some activities may be very distinct and can be segmented high up, while more subtle differences only appear deeper down. The concept of exploiting the temporal structure exhibited in human activities is similar to the *motion segments* in [Niebles *et al.* 2010] or the *micro-actions* in our previous work of Chapter 4.

In order to build up such a hierarchy of activities, we exploit the strong link between temporally adjacent observations in videos. Hence, activities are characterized 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.* The advantages of our approach are:

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. The dilemma between generalization capacity and precision of the model is naturally handled in the hierarchy. Nodes higher up in our hierarchical model are general and represent a broad variety of activities (*e.g.*, 'an object is moving'), whereas lower nodes only incorporate 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 discover the human activity concepts in such a hierarchical activity model.

5.3 Activity Discovery

Our approach is inspired by the principle of invariant or slowly varying features. Wiskott and Sejnowski [Wiskott and Sejnowski 2002] have proposed Slow Feature Analysis (SFA) as an unsupervised learning technique for continuous data streams, inspired by human learning capacities. Recently, Klampfl and Maass [Klampfl and Maass 2009] 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 and the transitions from one activity to the next are relatively short in comparison. For example, in the case of human action recognition, activities correspond to the

human actions, like walking or sitting, which are executed for a certain duration. The transitions between these activities are normally quite short without a clearcut boundary.

Given an image stream, $S = \{I_1, I_2, \ldots, I_T\}$ of T images, $I_t \in \mathbb{R}^{n \times m}$, each image I_t is represented by a D-dimensional feature vector $\boldsymbol{f}_t \in \mathbb{R}^D$. As the applications in Chapter 6 will show, the image representation is not critical and different feature types can be used.

5.3.1 Temporal Data Segmentation

In the segmentation step, the goal is to split the data stream into its composing activities. A broader set of activities is partitioned into temporally distinct 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:

$$\min J_{SFA} = \min \mathbb{E}_t(\Delta \boldsymbol{z}_t), \text{ where } \Delta \boldsymbol{z}_t = ||\boldsymbol{z}_t - \boldsymbol{z}_{t-1}||^2. \tag{5.1}$$

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 $\boldsymbol{z} = \boldsymbol{w}^\mathsf{T}\boldsymbol{y}$, it can be shown [Wiskott 2003] that the objective becomes

$$\min J_{SFA}(\boldsymbol{w}) := \frac{\boldsymbol{w}^\mathsf{T}\dot{\boldsymbol{D}}\boldsymbol{w}}{\boldsymbol{w}^\mathsf{T}\boldsymbol{D}\boldsymbol{w}}, \tag{5.2}$$

where

$$\boldsymbol{D} = \mathbb{E}_t(\boldsymbol{y_t}\boldsymbol{y_t}^\mathsf{T}) \tag{5.3}$$

is the covariance matrix of the data and

$$\dot{\boldsymbol{D}} = \mathbb{E}_t\big((\boldsymbol{y_t} - \boldsymbol{y}_{t-1})(\boldsymbol{y_t} - \boldsymbol{y}_{t-1})^\mathsf{T}\big) \tag{5.4}$$

the covariance matrix of the temporal differences. The weight vectors $\boldsymbol{w}$ which minimize Equation (5.2) are the solutions to the generalized eigenvalue problem

$$\dot{\mathbf{D}}\boldsymbol{w} = \lambda \boldsymbol{D}\boldsymbol{w}. \tag{5.5}$$

The slowest varying components in $\boldsymbol{y}$ are their projections onto the eigenvectors $\boldsymbol{w}$ associated to the smallest eigenvalues λ [Wiskott 2003]. We select the d_{SFA} slowest components to establish the subspace of slow features.

Clustering. In the SFA subspace, distinct activities are discriminatively mapped to distinct high density regions with sparse transitions [Klampfl and Maass 2009]. Hence, we choose to estimate a Gaussian Mixture Model (GMM) in order to find these high density regions (*i.e.* clusters) and assign the data-points to the activity clusters. By means of expectation maximization (EM) [McLachlan and Krishnan 1997], the Gaussian mixtures are iteratively refined and adapted to the data. Initialization is done with *k-means*.

Once the EM-algorithm converged, the cluster index to a data point is determined by the mixture component with maximal posterior probability [Bishop 2007]. Since the desired number of clusters is not known a priori, a sweep over k is performed and the model accuracy *i.e.* the overall log-likelihood $\ell(k, \mathcal{M}_k)$ as the sum over all data-points of the posterior probabilities given the model $\mathcal{M}_k$ is calculated. The relative difference $d(k)$ as a function of the number of clusters k, *i.e.*

$$d(k) = \frac{\ell(k, \mathcal{M}_k) - \ell(k-1, \mathcal{M}_{k-1})}{\ell(k+1, \mathcal{M}_{k+1}) - \ell(k, \mathcal{M}_k)}, \quad \text{with} \quad k \geq 2. \tag{5.6}$$

characterizes the curvature of $\ell(k, M_k)$. Intuitively this quantifies how much relative gain in likelihood each additional cluster yields. We then select

$$k^\star = \arg\max_k (d(k)) \tag{5.7}$$

as the optimal number of clusters, *i.e.* the number of clusters with the maximal likelihood gain[1]. Once k is set, every cluster determines a detected activity, and the data is assigned accordingly. A post-processing step ensures temporal smoothness and discards very short sequences.

5.3.2 Building the Activity Hierarchy

The data segmentation procedure explained above is applied recursively on the data. In the first step, we split according to the most dominant (slowest) cues in

[1] Indeed, in our case, the exact selection of k is not very critical. As seen subsequently, the model is further refined in a hierarchical manner, therefore activities that do not appear at one clustering step will be modeled further down in the hierarchy. This fact is actually one of the advantages of our hierarchical modelling procedure.

the entire data-stream, and a number of subsets (activity concepts) result from the procedure. Each of these concepts is further analyzed in order to create a hierarchical model. If necessary, the segmentation process is repeated for each obtained subset. As fewer data is now analyzed, discriminative components that were not apparent in the previous subspace may now appear. This is encouraged since we keep the dimensionality $d_{SFA} \ll D$ of the SFA subspace fixed across the entire hierarchy.

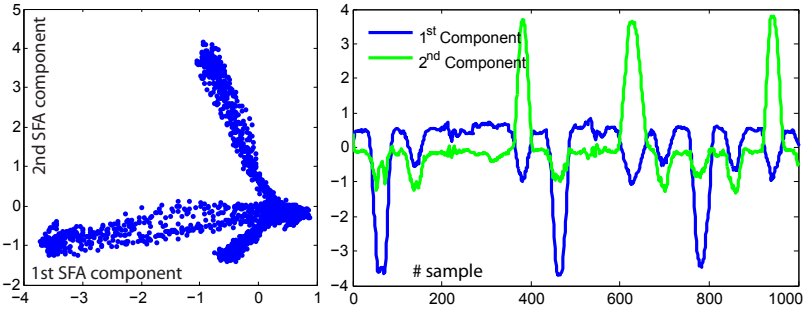
Each (sub-) set of segmented activity data refers to a node in the hierarchical model, as already sketched in Figure 5.2. At high levels, the established nodes T_i^j (node i on level j) contain very broad activity concepts while at lower levels in the hierarchy, specific actions are found.

Basic activities. The decision whether or not a node is further refined is based on the representation of the corresponding data in the SFA space. The data is projected so that the average distance between consecutive samples is minimized, *c.f.* Equation (5.1). If the distances are approximately equal across the whole sequence, the data is well described by its slowest components [Wiskott and Sejnowski 2002]. 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, such that the unit variance constraint is fulfilled. This case is consistent with the assumption of [Klampfl and Maass 2009], hence, splitting the data is stimulated.

As a simple measure of data compactness, we use the median of distances between consecutive samples in the projected SFA space. This median measure 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.

5.3.3 Illustration

To get an intuition of the processing, we now discuss our activity detection technique with respect to the human activity dataset introduced by Turaga *et al.* [Turaga *et al.* 2009] and show how our results compare to theirs. We use silhouette data from two views as provided by the authors, and encode them by applying a signed distance transform to each silhouette. In line with the feature



(a) Subspace *(b)* Temporal variation

***Figure 5.3**: The data projections (a) and the temporal evolution of the slowest components (b) characterize the discriminative SFA subspace.*

encodings in previous chapters, the distance transform is bounded to arbitrary maximal and minimal values, and reduces the sensitivity to small modifications at the edges when comparing two silhouettes. Finally, these silhouettes are downscaled to a fixed size and reshaped to a feature vector. The dataset includes five actions (*throw, bend, squat, bat, pick phone*), the all start and end with an idle upright pose. Each of action is repeated ten times at different execution speeds. We randomly permute the actions and the repetitions in order to form the input video. The goal is to automatically split this long activity video into its composing actions.

Following the procedure of Section 5.3.1, we project the activity data in to the subspace of slow features. In Figure 5.3(a) the resulting manifold is displayed. For visualization reasons, we chose $d_{SFA} = 2$. In Figure 5.3(b), the temporal evolution of the two slowest components is plotted. From the orthogonal behavior of the two slowest components, the discriminative characteristics of the SFA subspace is verified.

In Figure 5.4, the first two dimensions of the clustered SFA subspace ($d_{SFA} = 3$) are displayed. This manifold is obtained at the root node, where all five actions are included. The sketched hierarchy shows that four basic activities are extracted at the first split. The pink node is subdivided further, yielding two more basic activities. In Figure 5.5 the stopping criterion is verified. The empirical distributions of distances $\Delta \boldsymbol{z}_t$ and their medians are shown. For

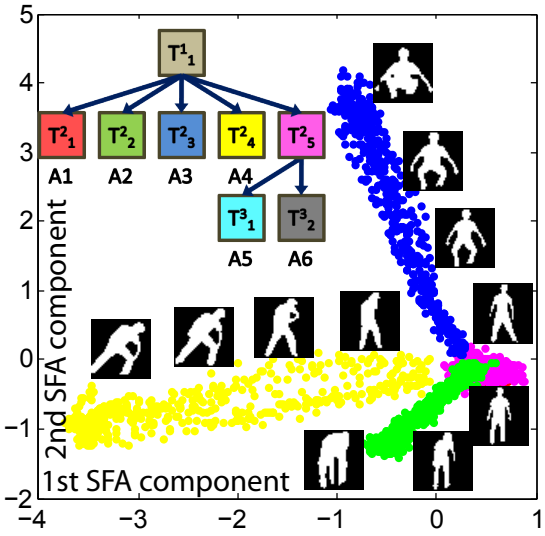


***Figure 5.4**: Illustration of the hierarchical activity discovery procedure. In the first three-dimensional SFA subspace, five activities are segmented. One of them is further refined (*c.f. *Figure 5.5). The clusters are indicated with color in the subspace, and the corresponding silhouettes are displayed. For visualization reasons, only two of the analyzed three SFA dimensions are shown.*

nodes T_1^1 and T_5^2, the shift of the mode towards the origin suggests to further split these nodes.

We automatically discovered six basic activities ($A1 - A6$). The samples that were filtered out during clustering (short sequences and outliers) are collected in $A0$. From the results reported in Figure 5.6, one can notice that activities $A1 - A5$ perfectly match the five ground-truth actions as defined by Turaga *et al.* [Turaga *et al.* 2009]. $A6$ corresponds to standing still, as observed at the beginning and the end of each action, but not annotated in the ground-truth. The confusion matrix in Figure 5.6(a) is obtained from the compositions of the ground truth snippets following the evaluation of [Turaga *et al.* 2009]. We clearly outperform their results (100% *vs.* 86% accuracy). The proportion of the discovered activities with respect to the total number of frames is reported in brackets and indicated by the field coloring. Since standing still is not included in the ground-truth annotation, this difference obviously lowers

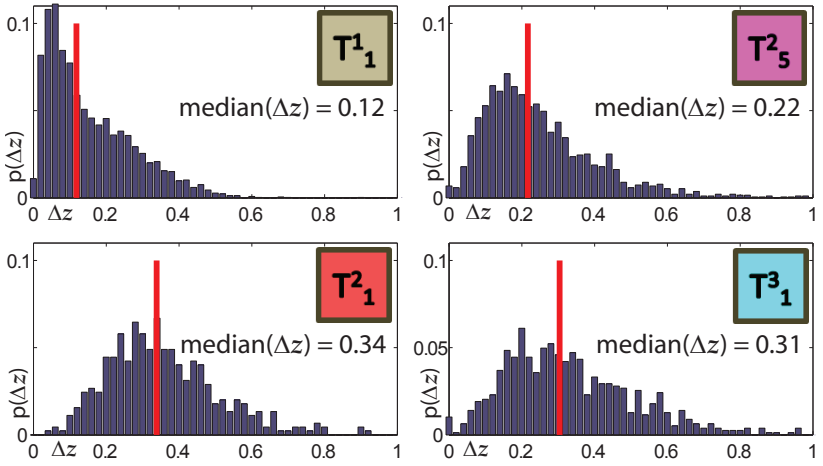


***Figure 5.5**: Demonstration of the splitting criterion based on the distribution of distances between consecutive samples in the SFA space. For basic activities, the median is higher, and they are not further segmented, while nodes T_1^1 and T_5^2 are subdivided (*c.f. *Figure 5.4).*

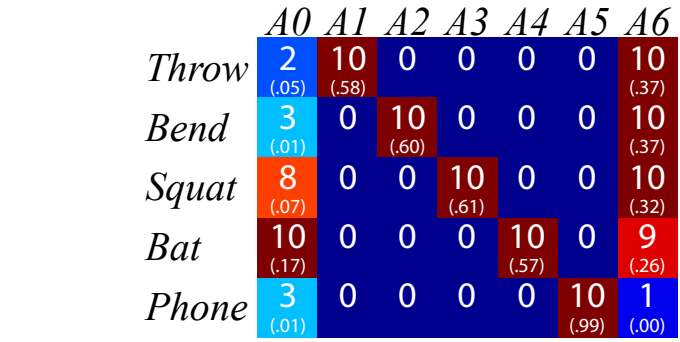
the values. In Figure 5.6(b) the temporal evolution of discovered and ground truth activities are depicted for half of the sequence. Again, the observation is that the our activity assignments are correct in every case, but sometimes incomplete or too short, due to the intermediate standing still.

5.3.4 Data Modeling

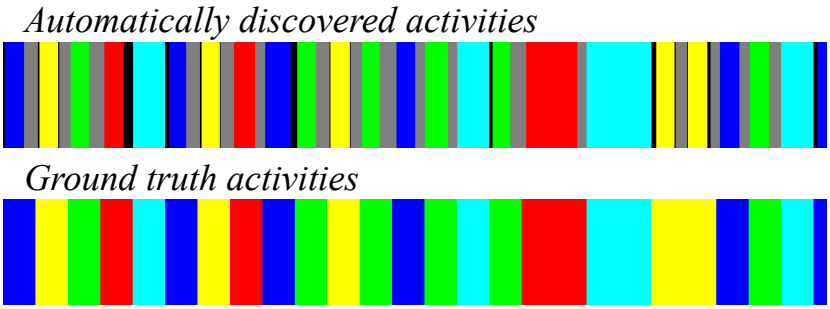
As we want to use the hierarchy to classify the activities in previously unseen videos, the data underlying each of its nodes is additionally modeled with respect to shape and dynamics. Biological studies on human motion perception suggest that motion analysis is performed from sequences of appearance snapshots [Lange and Lappe 2006]. Taking this into account, we create an extended feature vector

$$\boldsymbol{v}_t = (\boldsymbol{f}_t, \boldsymbol{f}_{t-1}, \dots \boldsymbol{f}_{t-n})^\mathsf{T} \tag{5.8}$$

as the concatenation of the last n feature representations, like proposed in [Urtasun *et al.* 2006b]. We model the zero-mean feature vector $\boldsymbol{x}_t = \boldsymbol{v}_t - \mathbb{E}_t(\boldsymbol{v}_t)$



(a) Confusion matrix



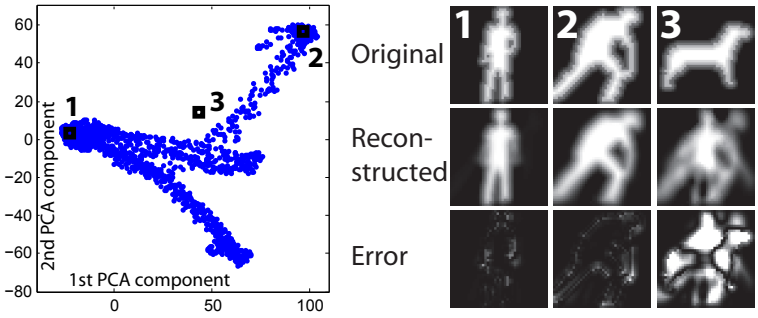
(b) Temporal activity segmentation

Figure 5.6*: (a) Automatically discovered basic activities ($A0 - A6$) vs. the ground truth, (b) Color coded labeling for the discovered and the ground truth actions (see text for details).*

by means of Principal Component Analysis (PCA). To show the analogies with the SFA formulation above, we briefly review this technique.

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 [Bishop 2007], *i.e.*

$$\max J_{PCA}(\boldsymbol{w}_p) := \mathrm{Var}_t(\boldsymbol{w}_p^\mathsf{T}\boldsymbol{x}_t) = \boldsymbol{w}_p^\mathsf{T}\boldsymbol{C}\boldsymbol{w}_p \tag{5.9}$$



(a) Subspace *(b)* Illustration

***Figure 5.7**: (a) Two principal components in the PCA subspace. (b) Two training images and one outlier projected and reconstructed.*

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:

$$e = \mathbb{E}_t\big((\boldsymbol{x}_t - \boldsymbol{x}_t^*)^2\big), \quad \text{where} \quad \boldsymbol{x}_t^* = \sum_{i=1}^{d} a_{t,i} \boldsymbol{w}_{p,i}. \tag{5.10}$$

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 Equation (5.9) is obtained by solving the eigenvalue problem

$$\boldsymbol{w}_p = \lambda_p \boldsymbol{C} \boldsymbol{w}_p. \tag{5.11}$$

The eigenvectors $\boldsymbol{w}_p$ that correspond to the d largest eigenvalues λ_p are selected as projection basis. In Figure 5.7(a), the two dimensional PCA manifold of the activity data at the root node T_1^1 is shown. As seen from Figure 5.7(b), this model represents well the training data, but has a high reconstruction error for an unfamiliar shape.

Starting from the same dataset, Figure 5.3(a) and Figure 5.7(a) show the two subspaces obtained with SFA and PCA, respectively. This verifies that, even though the formulations of the two techniques are similar, PCA creates a generative data model while SFA encodes differences and delivers the desired discriminative characterization.

Hierarchical Model The data in each node T_i^j in the activity hierarchy is represented with the model M_i^j in a PCA space with a fixed number of dimensions $d_{PCA} \ll D$. As seen, when moving down in the hierarchy, the data in each node describes more specific activity concepts. Likewise, the models naturally are more general at the top of the hierarchy and more precise at leaf nodes, as sketched in Figure 5.2. At the leaf nodes, each basic activity A is described by model M_A. As we do not assume the data to be free from abnormal situations and noise, we exclude the samples with highest reconstruction error at each step of activity refinement in the hierarchy.

5.4 Analysis of Unseen Data

We now show how the hierarchical model enables the efficient detection of known activities and abnormal situations that may occur in unseen data.

5.4.1 Activity Detection

Starting from an unseen sequence of images, we first run the same feature extraction procedure as proposed in Equation 5.8 to obtain the image features $\boldsymbol{x}'$. From the training, we dispose of a set of basic activities $\mathcal{A}$ and the task is to identify $A \in \mathcal{A}$, which best explains the new observations. To this end, $\boldsymbol{x}'$ is projected into the PCA subspaces that characterize the basic activities, and the reconstruction errors are calculated. The leaf node model M_A with be lowest reconstruction error e_{M_A} determines the discovered activity

$$A^\star = \arg\min_A e_{M_A}(\boldsymbol{x}'), \quad \text{where} \quad A \in \mathcal{A}. \tag{5.12}$$

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

Simultaneous target localization and activity detection. In certain applications, only a sub-region of the entire scene might be considered. For example, if the actions of a person are analyzed, the features will only describe this person but not the surroundings. In order to correctly detect the performed activity, this sub-region needs to be localized correctly. We therefore opt to jointly solve the tasks of target localization and activity detection and integrate

the search for an optimal location in the previous formulation for activity detection. At various image locations ρ (including scale), the reconstruction error $e_{M_A}(\boldsymbol{x}'|\rho)$ is determined for activity A. If we evaluate multiple activities, the optimal location and activity are found simultaneously, *i.e.*,

$$(\rho^\star, A^\star) = \arg\min_{A,\rho} e_{M_A}(\boldsymbol{x}'|\rho). \text{ where } A \in \mathcal{A}. \tag{5.13}$$

For efficiency reasons and since temporal consistency is assumed, only the local neighborhood of ρ^*_{t-1} (the location at the previous timestep) is scanned. This technique is usually referred to as *tracking*.

5.4.2 Exploiting the Hierarchy

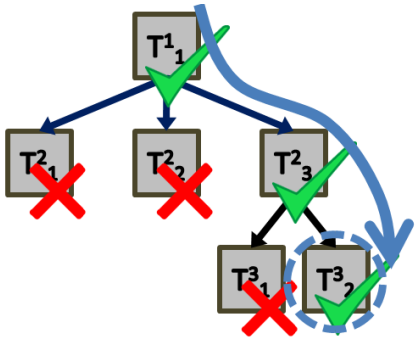
We now show how the hierarchical model paves the way for a more sophisticated and efficient analysis of unseen data. As seen previously, the hierarchy consists of a set of more general and more specific human activity models. Since each node (except the leafs, of course) is connected to its more specific sub-nodes, we can apply the anomaly reasoning in a disjunctive hierarchy, as proposed in [Weinshall *et al.* 2012]. To this end, we first need to determine if an observation is well described by a certain node in the hierarchy.

A node T_i^j with model M_i^j is considered *active* for an observation $\boldsymbol{x}'$ based on its normalized reconstruction error:

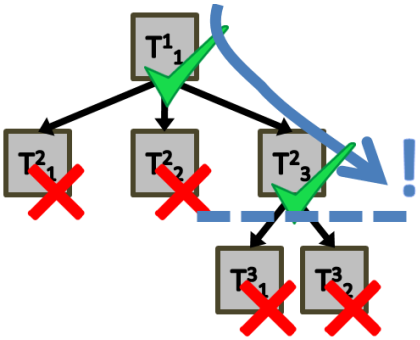
$$\text{active}(T_i^j) = \begin{cases} 1 & \text{if } \dfrac{e_{M_i^j}(\boldsymbol{x}') - \mu_{M_i^j}}{\sigma_{M_i^j}} < \theta \\ 0 & \text{otherwise} \end{cases}, \tag{5.14}$$

where $\mu_{M_i^j}$ and $\sigma_{M_i^j}$ are respectively the mean and the standard deviation of the reconstruction error for model M_i^j, obtained from the training data. θ is a user-defined threshold.

To respect the hierarchical arrangement of activity nodes, each observation is propagated from the root node to the leaves as sketched in Figure 5.8(a). Only sub-nodes of active nodes need to be considered, which increases the efficiency. As long as the observations are according to expectations captured in the model, there is always a leaf node (*i.e.* basic activity) which is able to explain the data.



(a) Valid activity



(b) Abnormal at level 3

Figure 5.8*: Use of the hierarchical model for the interpretation of unseen data. (a) A known activity is detected for an active leaf node. (b) A reasoning on abnormal conditions in the hierarchy is deduced from active and inactive nodes on different levels.*

If a more general node validates the observation, but none of its more specific sub-nodes does, then this signals an abnormal activity (Figure 5.8(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. While abnormalities detected high up in the hierarchy reveal severe deviations from the learned activity model, observations that are identified as abnormal close to the leaves exhibit only very subtle abnormal behavior. This will be experimentally verified in the next section.

5.5 Experiments

In this section we show how the proposed hierarchical activity model performs for the discovery and interpretation of human behavior in the same indoor scenario as already presented in Section 4.4. Further applications to different tasks and scene types with different feature configurations are described in Chapter 6.

5.5.1 Experimental Setup

Dataset For the human activity experiments, we use the dataset introduced in Section 4.4. The videos consist of images of 640×480 pixels, recorded at 15 fps. We use the image sequence *seq1* ($7,100$ frames) with different normal daily activities to train the model. The evaluation is carried out on the test sequence *seq2* ($1,030$ frames) that also contains abnormal events such as a fall.

Features We utilize the exact same image features as in Section 4.2.2, *i.e.* the distance transformed human silhouettes. For motion encoding, $n = 5$ last frames are concatenated.

Parameters. At training, an initial noise reduction step is applied to keep 95% of the data variance in each node. Subsequently, SFA and PCA subspaces are modeled with $d_{SFA} = 3$ and $d_{PCA} = 3$ dimensions. At test, the threshold $\theta = 3$ is applied for hierarchical reasoning. Different parameter choices sets did not change the results significantly.

Runtime. Due to the low complexity of the distance computation, the analysis of unseen data is very efficient. On a standard PC, our current MATLAB implementation runs at more than 12 frames per second. If a very accurate target localization is required, the exhaustive search procedure can slow down the evaluation up to a factor of 10. Model building takes in the order of a few minutes for our cases.

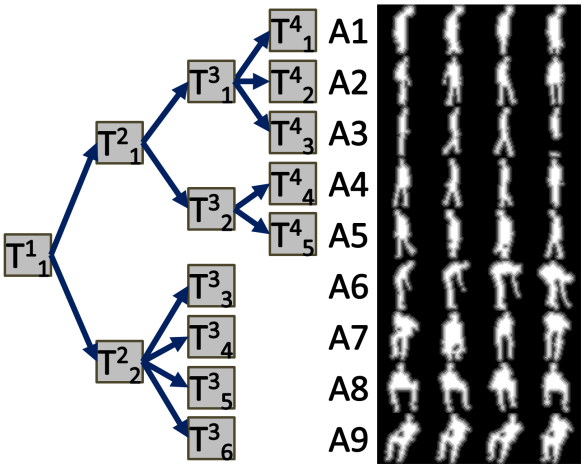


***Figure 5.9**: Experiments (1): From training, nine basic activities emerged im the the automatically learned hierarchy.*

5.5.2 Discovered Activities

The hierarchical model obtained form analyzing *seq1* is visualized in Figure 5.9. For each leaf node activity, some silhouettes are shown that constitute this activity. The hierarchy nicely encodes the different aspects of behavior observed in this video. At the hightest level, it distinguishes between upright poses in T_1^2 and all the other poses in T_2^2. At lower levels in the hierarchy, different actions are segmented or walking rightwards (T_1^3) is separated from walking to the left (T_2^3). Most of the discovered basic activities $A1 - A9$ have a unique semantic interpretation which can be annotated with little effort, as done in Table 5.1. Hence, meaningful human activities are discovered automatically. Only $A5$ seems to mix occluded and non-occluded leftwards walking. Note that this is achieved despite the noisy silhouette features sometimes containing holes or gaps.

5.5.3 Runtime Analysis and Abnormal Event Detection

The model is applied to the test sequence *seq2* and the unseen observations are assigned to previously learned basic activities or abnormal situations are de-

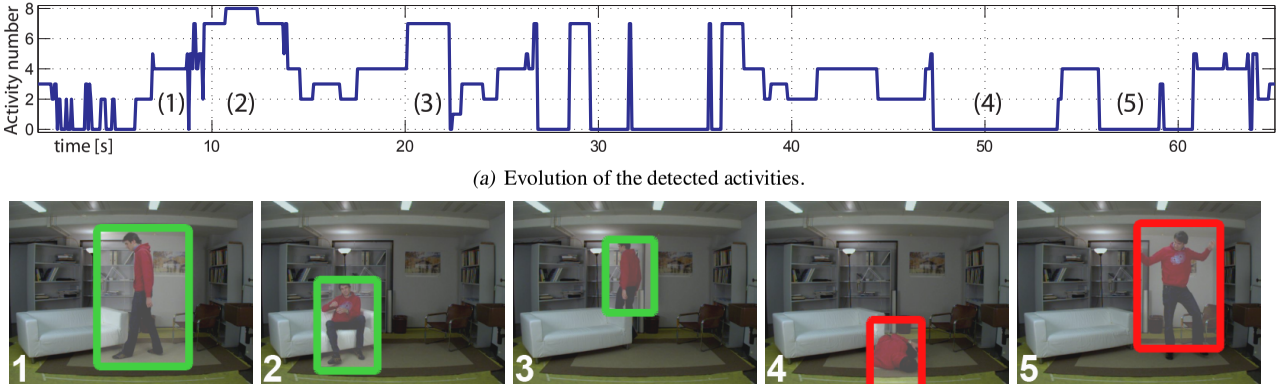
Basic activity	Human interpretation
$A1$	Occluded walking to the right
$A2$	Turning from frontal walking towards the right
$A3$	Walking to the right
$A4$	Turning from frontal walking towards the left
$A5$	Occluded and non-occluded walking towards the left
$A6$	Picking up an item
$A7$	Sitting down
$A8$	Sitting on the couch (sitting position 1)
$A9$	Sitting on the armchair (sitting position 2)

***Table 5.1**: Manual human annotation of the nine discovered basic activities.*

tected. The observed person is tracked throughout the video and the matching activity is determined simultaneously. The plot in Figure 5.10(a) characterizes the evolution of the detected basic activity over time. $A0$ groups the outliers, some of them are manually annotated. In Figure 5.10(b) some selected frames of the test sequence are displayed, they show three normal situations and two detected anomalies.

We quantitatively compare the overall performance of the proposed technique to the previous results of Chapter 5. The recall-precision curve is obtained by sweeping the parameter θ (see Equation 5.14) and is displayed in Figure 5.11. We group the outliers detected at any level in the hierarchy and detect them as abnormal events. It appears that we outperform our previous technique. In particular the recall is increased from 68% to approximately 83% at 99% precision.

Due to the integration of temporal relations in the model building process, the discovered activities turn out to be more accurately segmented and well interpretable, compared to the previously used *k-means* clustering. This enables a precise modeling of the essential aspects of the human activities even with a simple PCA model, and allows for accurate abnormal event detection during runtime.



(a) Evolution of the detected activities.

(b) Snapshot frames of the test sequence.

Figure 5.10: *Experiments (2): In (a), the re-detected activities in the test video are reported over time with the activity number as in Figure 5.9. A0 corresponds to an anomaly. The instants of frames* $1-5$ *are indicated. In (b) these selected frames of the test video are shown, illustrating the detected normal and abnormal situations: 1. Walk leftwards, 2. Sit, 3. Walk occluded, 4. Fall, 5. Wave heavily.*

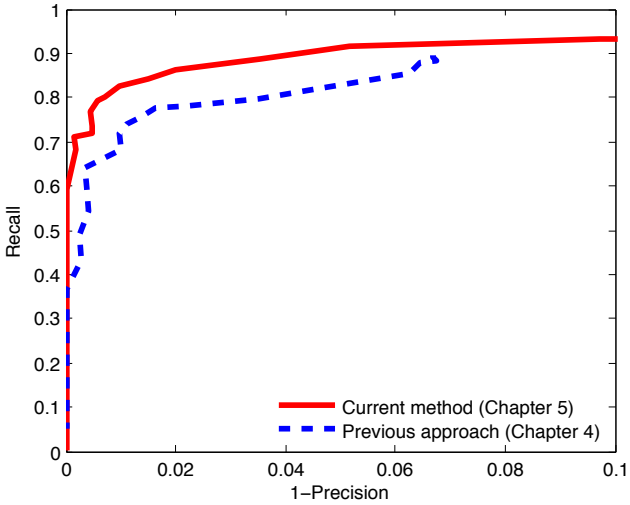


***Figure 5.11**: Experiments (3): Recall-precision curve for the abnormal event detection task. The manual annotations of the test video are compared to the automatically detected anomalies. The technique introduced here outperforms the previous approach of Chapter 4.*

5.6 Conclusions

In this chapter, we presented a data-driven approach to activity segmentation that exploits the temporal relations in video sequences. The small changes from frame to frame are examined with slow feature analysis, in order to automatically represent the data in a meaningful hierarchy. We have shown how this model is applied to unseen videos and that the hierarchy can be used to explain the observations. Due to two linear techniques of low computational complexity, we are able to efficiently detect normal and abnormal activities. Finally, qualitative and quantitative results demonstrate the validity of our technique.

6

Applications beyond Human Activity Analysis

6.1 Introduction

Many of the approaches to the unsupervised analysis of scenes or (human) behavior and abnormal event detection are specifically designed for a particular application, for example by relying on sophisticated feature types or scenario-specific assumptions. This limits the use beyond the application, they are proposed for. In this category falls the approach of Chapter 2, which is exclusively geared towards the interpretation of human motion.

In this chapter we show how our previously introduced techniques can be applied to different surveillance scenarios and extract useful information. In fact, both techniques proposed in Chapter 4 and Chapter 5 are widely applicable and not restricted to human silhouette features. Here, we only show applications of the activity analysis of Chapter 5. As we have seen in the experiments of Section 5.5, it reports superior results due to the incorporation of the temporal characteristics in the model building process.

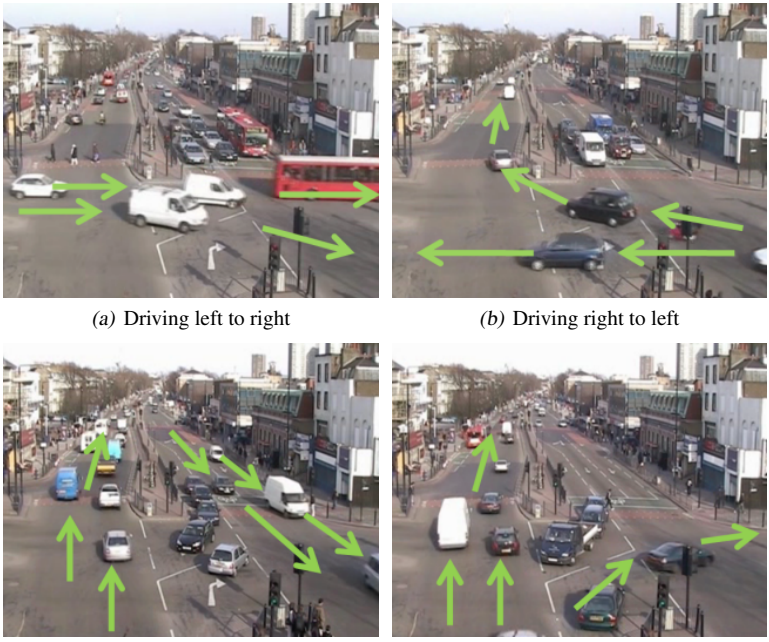
In the following, we present two experiments that aim at the interpretation of street scenes (Section 6.2) and one experiment that deals with the analysis of webcam image streams with low, possibly variable frame rate (Section 6.3). Subsequently in Section 6.4, we show how our technique is suited to analyze industrial workflows if we introduce additional, application-specific constraints. Such a workflow model can be used to interpret previously unseen videos, as demonstrated in Section 6.5.

The contents of this chapter were published in [Nater *et al.* 2011a] and [Nater *et al.* 2011b].

6.2 Traffic Scene Analysis

In two experiements, we show how the technique of Section 5 is applied for the surveillance of traffic scenes. Normal behavior and abnormal events are automatically identified.

6.2.1 Experiment 1: QMU Junction



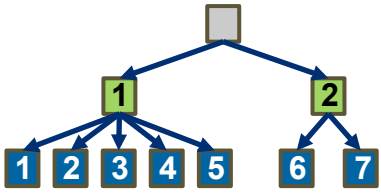
(a) Driving left to right

(b) Driving right to left

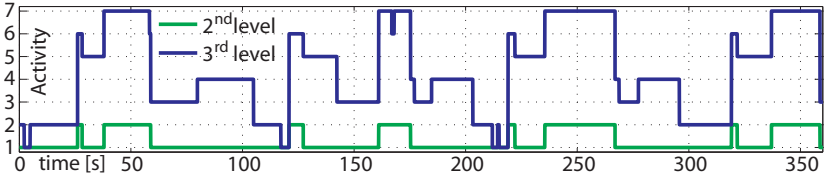
(c) Both lanes driving straight

(d) Left lane driving straight and turning right

***Figure 6.1**: Selection of detected basic activities characterizing the traffic flow on the street junction.*



(a) The first three levels in the tree



(b) Activities plotted over time for the levels corresponding to (a).

***Figure 6.2**: The hierarchical model for the QMU Junction consists of 29 activity nodes. The first three layers are scematically displayed in panel (a) and the activity membership is plotted over time in (b).*

In a first experiment, we use the data from [Hospedales *et al.* 2009], that shows a busy street junction with vehicles driving and turning in all directions. The footage is recorded during 1 hour at a frame-rate 25 fps and an image resolution of 360×288 pixels. This data has previously been used for learning spatio-temporal scene topics in [Hospedales *et al.* 2009].

Our technique of Section 5 works independently of the choice of features, hence we use the feature type that seems to be most suitable for the application. As the most characterizing scene behavior is vehicle and pedestrian motion, we choose to employ motion descriptors. Following the concept of [Veres *et al.* 2010], we calculate the motion grid across the entire scene based on local motion monitors of 18×18 pixel patches. To additionally encode the motion direction, a forgetting rate of 0.95 is applied. Training is done on $50,000$ images, the runtime evaluation takes into account all $90,000$ frames.

The learning procedure extracts 98 nodes of which 19 are basic activities at the leafs. Some of these discovered basic activities are depicted in Figure 6.1. As can be seen, the hierarchical analysis nicely groups co-occurring and interpretable traffic patterns in leaf nodes. Further basic activities for example summarize streets with only pedestrian motion, cars accelerating, and different

(a) Ambulance interfers

(b) Wrong driving direction

(c) Collision course

(d) Conflicting motion of minibus in the middle

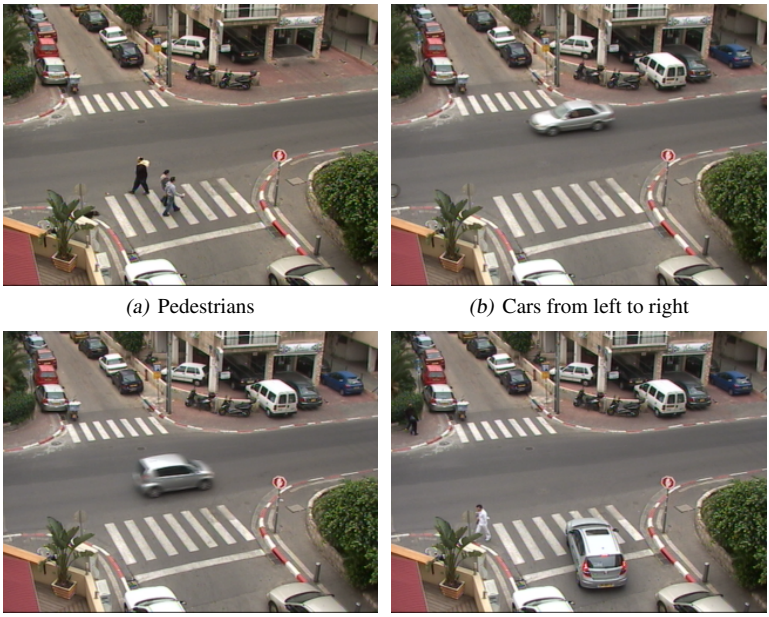
***Figure 6.3**: Detected anomalies that occured in the scene. Conflicting Image regions are shaded in red.*

turn configurations. In Figure 6.2 we show the obtained activity plotted over time, for the second and third level in the hierarchy, together with this part of the activity tree. Without enforcing any larger scale temporal relations, we discover pseudo-repeated patterns in the data that correspond to different phases in traffic light cycles. As successfully done for example in [Kuettel *et al.* 2010], these patterns can be additionally learned for the detection of irregular ordering of different familiar activities.

Applying the hierarchical model to unseen data, we can discover diverse irregular situations. Four such examples are depicted in Figure 6.3, the ambulance and the wrong driving direction have also been reported in [Hospedales *et al.* 2009]. Since we use a holistic scene descriptor, unseen configurations, or con-

flicting motions in different scene regions are also reported. This is for example useful to detect collision courses or unexpected vehicle motion. Among all the detected abnormal events, there are hardly any that have no plausible interpretation.

6.2.2 Experiment 2: HUJI Street Crossing



(a) Pedestrians *(b)* Cars from left to right

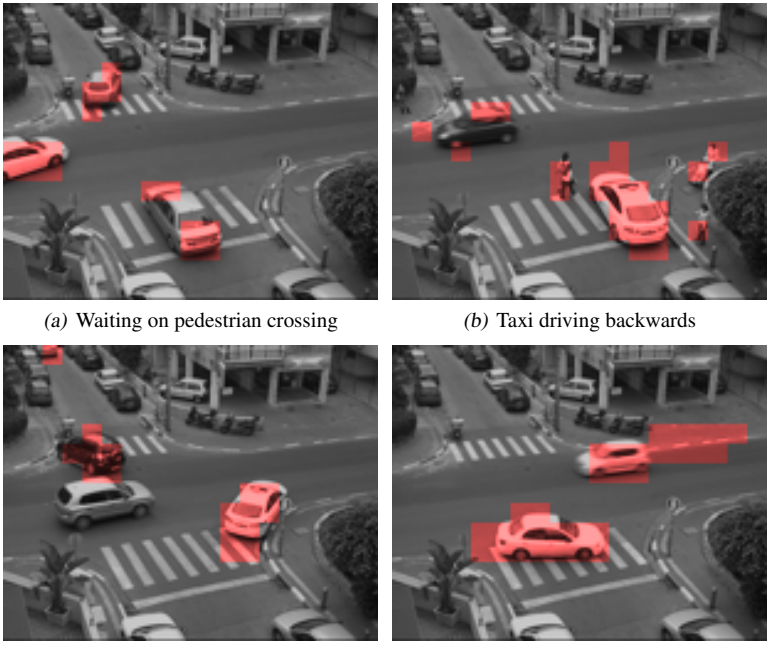
(c) Cars from right to left *(d)* Cars from bottom

***Figure 6.4**: HUJI crossing: Activities discovered during training, representing typical street motion.*

In a second experiment, we use the HUJI street crossing footage from [Hendel *et al.* 2010]. This 2 hour long video is recorded at 10 fps with frames of 320×240 pixels. We utilize the same motion descriptors as for the previous scene. The model is trained with the video of the first hour, and again, typical scene activities are discovered. The model consists of a tree with 16 nodes of which 11 are basic activities. Some of them are reported in Figure 6.4, others

show different car turns or combinations, such as cars driving from right to left and from left to right.

We apply the model to the second hour of recorded footage and a few detected abnormal events are depicted in Figure 6.5. These detections mainly relate to abnormal driving patterns. For example, in panel (c), one can nicely observe, that the silver SUV is normally driving on the street from left to right, whereas in this combination, the taxi's motion is inexplicable and hence reported as dangerous. In Figure 6.5, the observed driving direction in the region of the pedestrian crossing is also detected and reported as abnormal.



(a) Waiting on pedestrian crossing

(b) Taxi driving backwards

(c) Collision course

(d) Abnormal driving direction

***Figure 6.5**: HUJI crossing: Spotted and manually interpreted anomalies.*

6.3 Webcam Stream Analysis

Despite the fact that webcams typically have a small, often irregular framerate, the temporal relations between frames can still be expoited. They simply live in a different time-scale than the previous examples. In the case of webcams, activities do no longer correspond to vehicle or pedestrian motion, but reflect for example day-night changes, shadow motion, periods of busy traffic or empty streets. We show this on the example of the Times Square Webcam dataset of [Breitenstein *et al.* 2009b].

6.3.1 Dataset and Features

In the dataset of [Breitenstein *et al.* 2009b], images from a webcam overseeing Times Square in New York are recorded at low frame rate over a long period of time. The images have a resolution of 640×480 pixels, the frame-rate is approximately 0.3 fps and we dispose of recordings from 2 months. For our processing, we only use every 3^{rd} frame. We downsample the original color images to 24×32 pixel grayscale images and concatenate the rows to a vector. Due to the low frame rate, we do not include motion descriptors ($n = 1$).

6.3.2 Discovered Hierarchy

The hierarchical model is obtained with data from 17 days comprising approximately $150,000$ images. The discovered hierarchy has 39 nodes, thereof 26 basic activities at the leaves. In Figure 6.6, we display the tree-like structure for the first four levels, and show typical instances of some nodes, together with their human interpretation. Day-night changes turn out to be the most dominant cues, which are separated in the first step.

6.3.3 Abnormal Events

In Figure 6.7 we show nine illustrative abnormal events that are detected among more than $250,000$ evaluated frames. We detect similar anomalies as reported in [Breitenstein *et al.* 2009b], such as the first four cases of Figure 6.7. In addition, our method also reported many cases of incomplete and broken frames, camera failures, water on the lens and other salient situations.

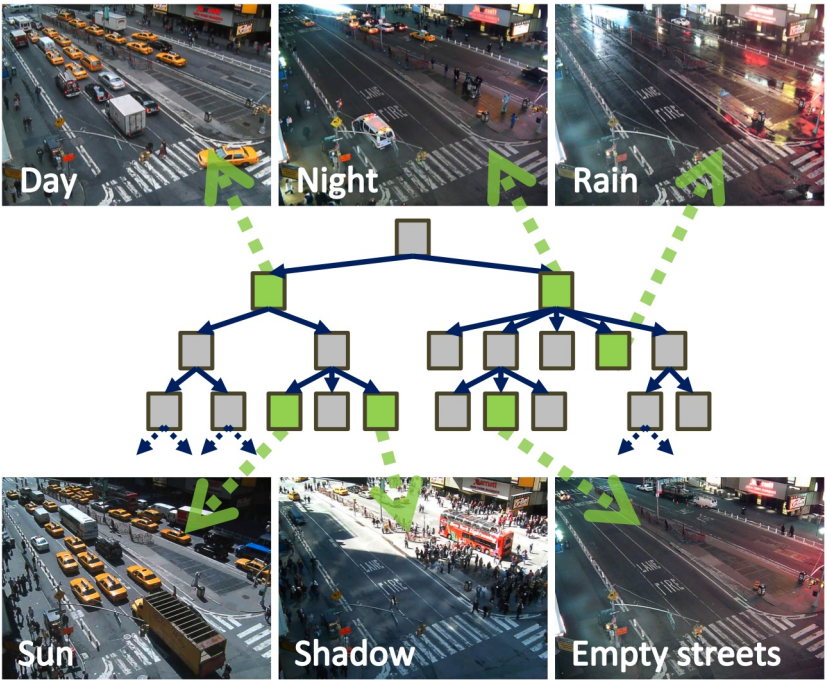


***Figure 6.6**: Part of the obtained tree with interpreted activities for the Times Square dataset*

.

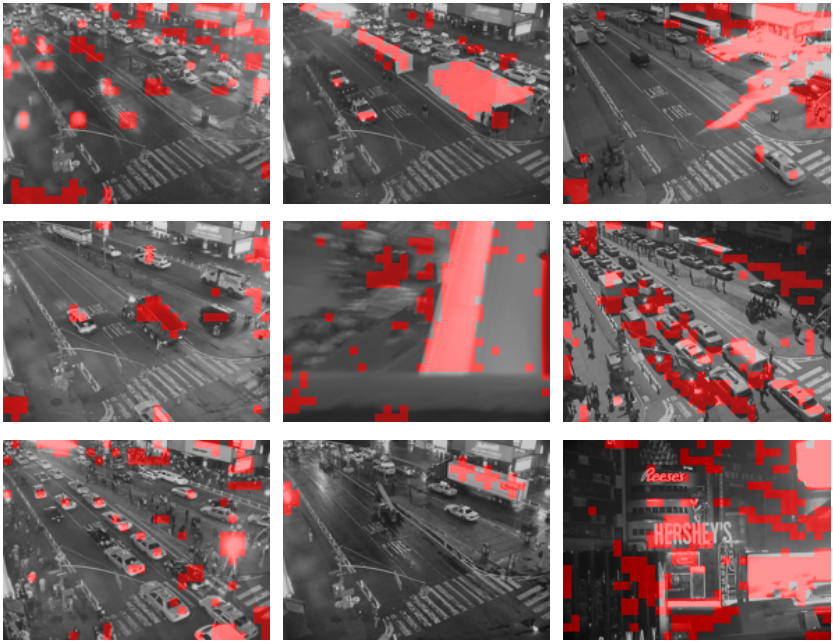


***Figure 6.7**: Automatically detected anomalies: Heavy rain on lens, festival tent for street festival, particular shadow shape; parked trucks for maintenance work, camera failure, jam with reflections; strong flashlight, truck parked in the background, camera moved.*

6.4 Task Discovery in Industrial Workflows

In this section, we aim at the unsupervised discovery of tasks that constitute a manufacturing workflow, whereas in the Section 6.5, we show how the workflow can be modeled in order to interpret previously unseen data and detect abnormal events.

6.4.1 Overview

Surveillance tasks are nowadays increasingly augmented with vision systems and smart algorithms to extract information or detect precise (abnormal) events. In this work, we focus on the interpretation and analysis of industrial scenarios. Hereby, several challenges must be overcome, such as unfavorable working conditions with dust, sparks or vibrations, cluttered background, diverse moving objects or heavy occlusion of the workers. Additionally, the workers look very similar, as they often wear utility uniforms. In this context, one issue is to monitor the smooth running of a workflow and detect any abnormal behavior. Deviations from the workflow may cause severe deterioration of the product quality or may raise safety or security hazards. Usually, the (normal) workflow has to be defined beforehand, which is done in an initial training phase with human intervention.

Relatively few work has been done for the analysis or automatic extraction of workflows. Recent medical applications use computer vision techniques to monitor surgical workflows [Blum *et al.* 2010, Padoy *et al.* 2009] in supervised settings. Due to the challenging conditions in industrial environments, sophisticated image processing methods, such as the detection and tracking of objects or persons are hardly applicable. Approaches which build on these techniques are very likely to fail in practice. Hence, in the setting of industrial workflow monitoring, Veres *et al.* [Veres *et al.* 2010] proposed to use a holistic scene representation. The main drawback of all these approaches however is their need for a manually pre-defined workflow model and annotated tasks. They can only monitor, but not discover workflows.

Based on the elaborations of Chapter 5, we propose a method to extract meaningful and interpretable workflows in an completely unsupervised manner. In order to overcome the involved challenges, we make use of clear assumptions

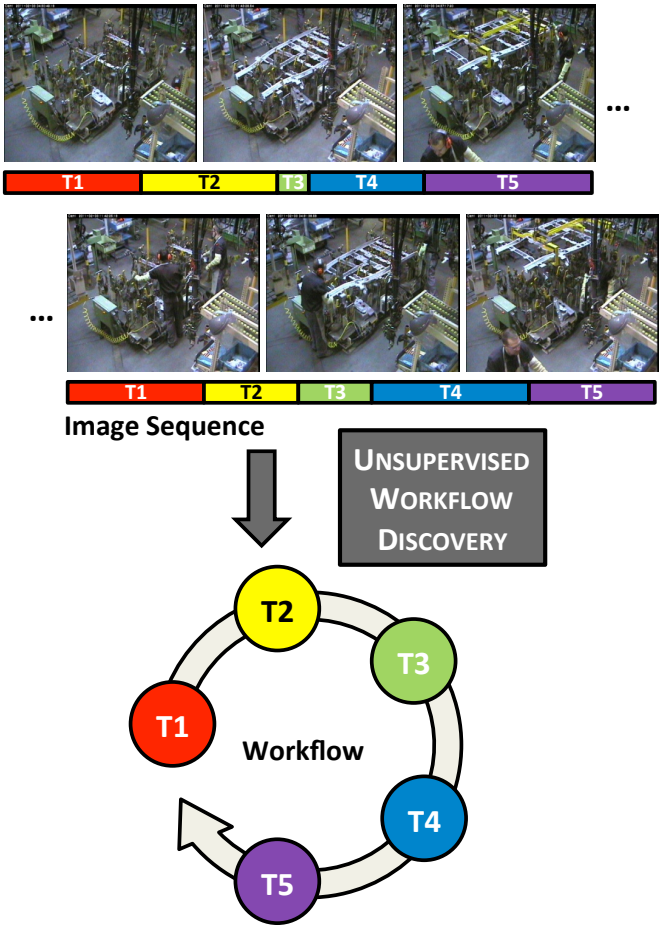


Figure 6.8: *In industrial environments, assembly tasks typically have a repeated cyclic structure. They are called workflows and consist of several tasks. The number of tasks as well as the segmentation is unknown. The goal of this work is to extract the workflow in an unsupervised manner and provide a simple yet effective analysis of industrial activity.*

that hold for industrial scenarios, such as the repeated structure of the workflow. To the best of our knowledge, we are the first to model workflows without any human intervention during the discovery process.

With our simple yet effective technique, we examine videos of an assembly line in a car manufacturing site. An example of such an industrial scene is depicted in Figure 6.8 which shows an extract of the car production process and the goal is to establish the assembly cycle in an unsupervised manner. The extracted workflows turn out to be consistent across different camera views and well interpretable, also compared to independent human annotation. In addition, in Section 6.5 we analyze several hours of video data in real-time, which allows us to interpret the workflow, and reason on different abnormal situations. In fact, the obtained statistics can be used in order to optimize the workflow and enable a safe running of the monitored assembly process.

6.4.2 Prerequisites

Goal. Given an image stream from a video camera, we aim to automatically discover the underlying workflow. No pre-segmentation of the image stream nor any other supervision is assumed to be available. Let us first define the following terms used:

Task: A task corresponds to a (physical) action, such as to pick up an object and place it somewhere.

Workflow: A workflow consists of a certain number of tasks and their transitions.

The goal of workflow discovery is to extract a number of N tasks T_n, with $n \in \{1, \dots N\}$ and N unknown, that represent the workflow observed in the scene.

Assumptions. As we aim for a widely applicable approach, we do not rely on explicitly modeling or recognizing humans, actions, or objects within the scene. Furthermore, we do not impose restrictions on the camera viewpoint. Yet, we have noticed some given factors that permit to set up assumptions concerning the nature of the workflow. They are described in the following.

Static camera: We assume that the workspace is monitored by a static camera.

Image sequence: We assume the image sequence to be temporally consistent, *i.e.* neighboring image frames are correlated and are likely to share a common task label.

Cyclic workflow: We assume the workflow to have cyclic layout, *i.e.* the tasks have always the same ordering and are repeated.

In other words, we are looking for a cyclic workflow observed by a video camera, as outlined in Figure 6.8. These assumptions are usually satisfied in industrial assembly lines, where parts or goods are manufactured or assembled systematically in an identical and repetitive manner. In fact, it is essential to produce in regular working cycles in order to maximize output while reducing defects and wastes.

6.4.3 Workflow Extraction Procedure

Our approach to the automatic discovery of a workflow makes use of the above assumptions and consists of (i) noise reduction for robust analysis, (ii) potential task spotting and (iii) temporal refinement. The individual steps are described in more detail in the following.

Noise reduction. The fact, that we are using a static camera, allows us to use the complete image and extract a holistic image representation. Given a sequence of images $\boldsymbol{x}_t \in \mathbb{R}^d$, we apply Principal Component Analysis (PCA) [Bishop 2007] on the zero-mean input feature vectors $\hat{\boldsymbol{x}}_t$. The data is projected onto its eigenvectors, and these projections, sorted with respect to the eigenvalues, span a new orthogonal space. In the first dimensions, maximal variance of the initial data is encoded, while dimensions with small eigenvalues most likely represent noise. We choose to select the $n_{PCA} \ll d$ first components in order to keep 80% of the total variance. $\boldsymbol{y}_t \in \mathbb{R}^{n_{PCA}}$ is the projection of $\hat{\boldsymbol{x}}_t$ onto these components.

Identification of potential tasks. In Chapter 5, we have shown that the temporal structure in image sequences provides a strong cue for learning representations. Following this approach, we first learn an embedding using Slow Feature Analysis (SFA) that explores the temporal dependencies in the data. Subsequently, we cluster the data in the obtained lowdimensional subspace.

Extraction of invariant signals. SFA [Wiskott and Sejnowski 2002] is a technique to automatically extract the invariant components in temporal signals. The output signal $\boldsymbol{z}_t$ of the SFA represents the slowest components in $\boldsymbol{y}_t$, as

seen in Section 5.3. Hence, we apply the same procedure to extract the slowest components from the original video signal and select the n_{SFA} slowest dimensions to span the SFA subspace.

For illustration, Figure 6.9 depicts the first four slow features over time for the industrial dataset that will be introduced in Section 6.4.4. The repeated workflow structure can be clearly observed.

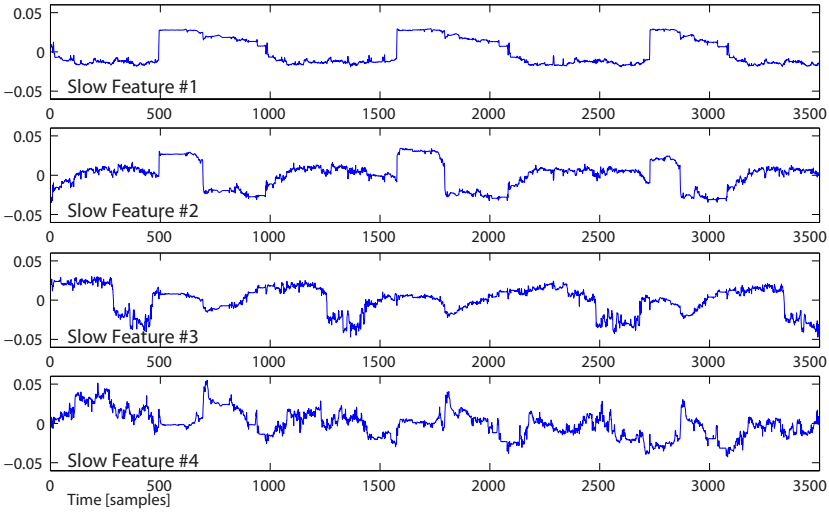


***Figure 6.9**: The four slowest features over time. The repetitive structure of the workflow appears in the first dimensions of the SFA subspace, whereas in higher dimensions, irregularities are encoded.*

The slow features are extracted in an unsupervised process and do not necessarily have to represent the desired cycles. It is possible, that slow features also encode variations in the scene that do not belong to the workflow we are looking for. This might be due to other overlapping workflows (*e.g.*, bringing goods to the workplace), variations on a longer period of time (*e.g.*, illumination changes), or other background motion. It has been recently shown in the medical community that a selection of the SFA components is feasible in order to focus on the task one is looking for [De Luca *et al.* 2011]. However, we did not observe such issues in our experiments.

Clustering. In the subspace of selected SFA components, the tasks appear as clusters of data-points. We choose to apply mean shift clustering [Comaniciu and Meer 2002] because of its robustness and its capacity to discover nonlinear cluster structures. Furthermore, we do not need to manually fix the number of clusters to extract.

Following the SFA subspace properties, we choose the bandwidth of the mean shift kernel as the expected temporal variations $\mathbb{E}_t(\Delta \boldsymbol{z}_t)$. A two-dimensional SFA embedding and the obtained cluster centers in black are shown in Figure 6.10.

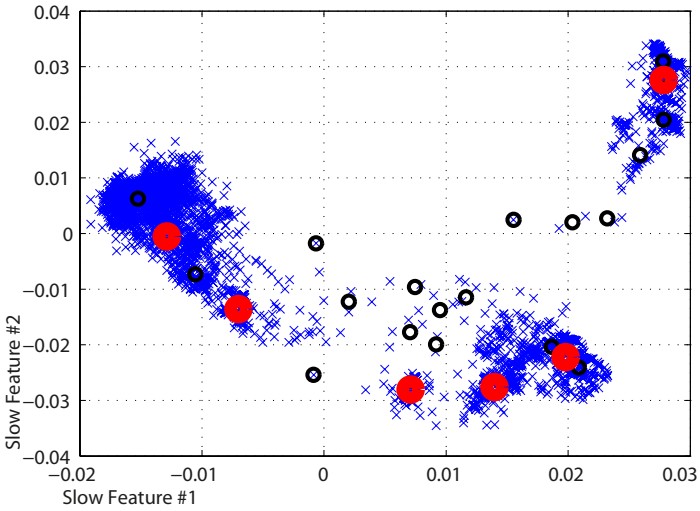


***Figure 6.10**: Mean shift clustering in two-dimensional SFA space, the initially detected 21 clusters (black) are refined to six final tasks (red) in the workflow.*

Temporal refinement. Like the original data, the measurements in SFA space are always affected with noise. Based on the assumption of temporal consistency and of a cyclic workflow, we apply the following refinement steps:

Task duration: Tasks are required have a certain duration. Therefore, very short tasks which only consist of a few images are considered as noise (outliers) and are removed. In practice, we eliminate all clusters which are shorter than 5 seconds.

Cyclic workflow: By analyzing the task transitions, a cyclic workflow is enforced. Tasks are merged if they jitter or if they yield splits in the workflow.

In more detail, two tasks T_i and T_j are said to jitter and are merged if

$$P(T_i|T_j) > \Theta \wedge P(T_j|T_i) > \Theta, \tag{6.1}$$

where $P(T_i|T_j)$ is the transition probability from T_j to T_i obtained from the clustered data. Θ is a small user defined threshold, we used $\Theta = 0.1$ for all experiments.

The assumption of a cyclic workflow implies a unique path, *i.e.*, from one task T_k only one dominant transition is allowed. Hence, we merge two tasks T_i and T_j if

$$P(T_i|T_k) > \Theta \wedge P(T_j|T_k) > \Theta. \tag{6.2}$$

To illustrate this procedure, exemplary task transition matrices before and after imposing the cyclic workflow layout are depicted in Figure 6.11. The centers of the finally emerged tasks (clusters in the SFA subspace) are marked red in Figure 6.10.

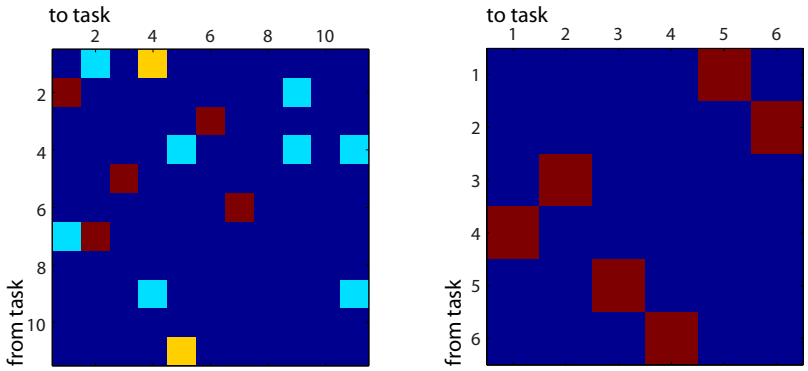


Figure 6.11*: Transition probabilities between the tasks: (a) after elimination of small (temporally short) tasks, (b) after imposing a cyclic workflow structure. The color encodes the (normalized) number of transitions that are observed from one task to another.*

Model selection. In many subspace problems, it is unclear how to select the optimal number of latent dimensions. We propose to estimate this model complexity from the resulting number of tasks.

If a small number of dimensions is chosen, only few clusters emerge and the model of the working cycle might be overly simple with very general tasks. On the other hand, if we select more dimensions, many detailed states are identified, but they might degenerate and not fulfill the cyclic workflow assumption. Hence, these states are merged during refinement, which results again in a small number of final tasks. This said, we sweep over dimensionalities and choose the subspace such that the number of tasks in the workflow is maximized. This intuition is verified in Table 6.1, where the number of clusters (tasks) are indicated before and after refinement. In this case, we select the SFA subspace to be 2-dimensional ($n_{SFA} = 2$) and the discovered workflow comprehends six tasks.

SFA dimensionality	1	**2**	3	4	5	6
# of initial clusters	8	**22**	39	84	135	170
# of long tasks	6	**11**	13	13	12	12
# of final cyclic tasks	3	**6**	6	5	4	4

***Table 6.1**: Number of initially detected clusters, and discovered tasks as a function of the SFA subspace dimensionality. In this example, a two-dimensional representation is selected.*

6.4.4 Experimental Setup

Dataset. For our experiments, we use the data which was recorded in the SCOVIS project.[1] The data is recorded in a car manufacturing facility and the sequences show close views of an assembly area. Two camera views are provided, the first one monitors the working cell from the side and the second one is mounted overhead. The RGB-colored frames have a resolution of 704×576 pixels and are recorded at a framerate of $18 - 25$ fps. For the side view camera, recordings were made for approximately 1.5 working days.

[1] www.scovis.eu, 3^{rd} SCOVIS industrial dataset (shared upon our request and publicly available soon).

Preprocessing. As input to our workflow analysis we convert the images to grayscale, downscale them by a factor of 8 (88×72) and finally reshape them to a 6336-dimensional feature vector. In all our experiments, we only analyze every 15^{th} frame.

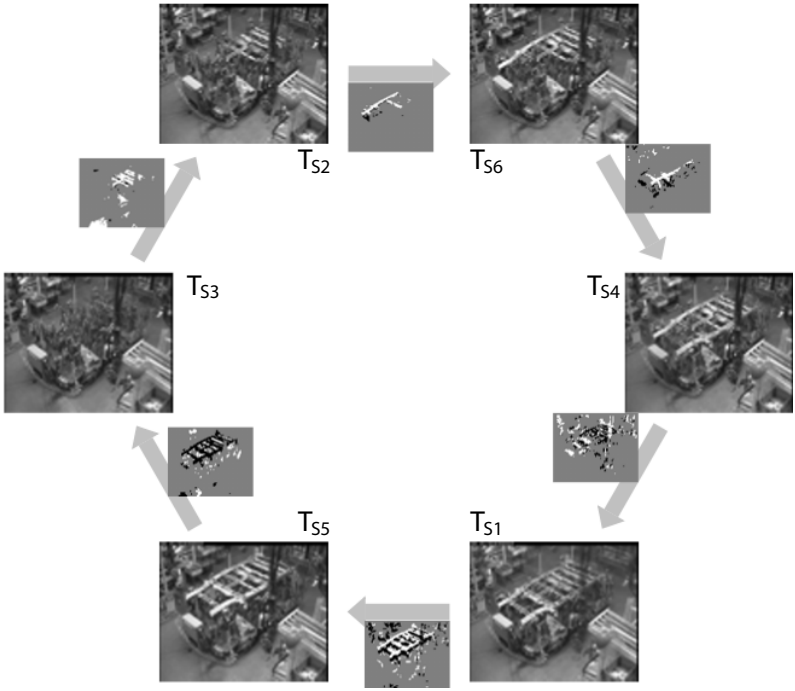


***Figure 6.12**: Automatically discovered cyclic workflow for side view. The tasks are indicated with their mean images and the transitions are shown.*

6.4.5 Discovered Workflows

We use the first hour of recordings for both camera views to apply our proposed automatic workflow discovery algorithm. The algorithm chooses in both cases a 2-dimensional SFA embedding. Details for the side view have been shown to illustrate the procedure in Section 6.4.3.

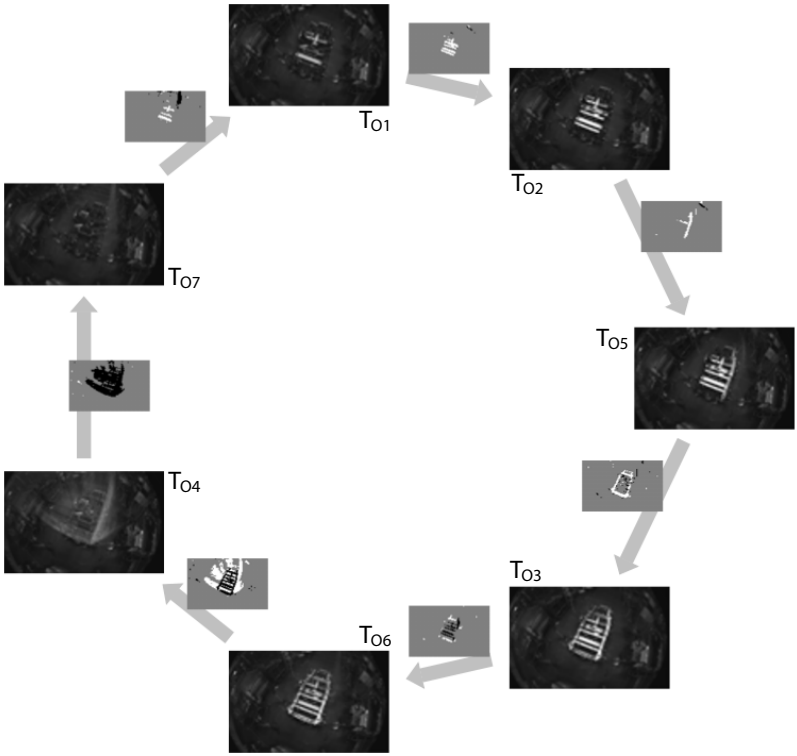


***Figure 6.13**: Automatically discovered cyclic workflow for the overhead view (camera mounted above the working area). The tasks are indicated with their mean images and the transitions are shown. Note the good correspondence of discovered tasks in the two views.*

The discovered tasks in the workflows for the side view and the overhead view are depicted in Figure 6.12 and Figure 6.13, respectively. Tasks are represented by their mean images and are connected with directed arrows. The small images next to the arrows depict the average variations of the image intensities from one task to the next. Each task is numbered with the according task index, and a manual interpretation for the tasks is given in Table 6.2.

For the side view, six tasks are established, whereas for the overhead view, seven tasks are found. It appears that the tasks correspond well between the two viewpoints. All tasks have its relative counterparts in the other view except

for T_{O1} and T_{O2}, which are merged in the side view to T_{S2}. This is probably caused by the fact that the assembly of the different small spare parts is not very distinctive in the side view. Therefore, a single task is discovered, whereas in the overhead view, this placement is split into two separate tasks.

For the side view, the temporal sequence of tasks is shown in Figure 6.14 for the training video. The cyclic behavior of the workflow is apparent, even though, due to the unsupervised clustering, the assignment of an index to a task is not necessarily in the order of the workflow. However, if required, this can easily be adapted by switching the task indices.

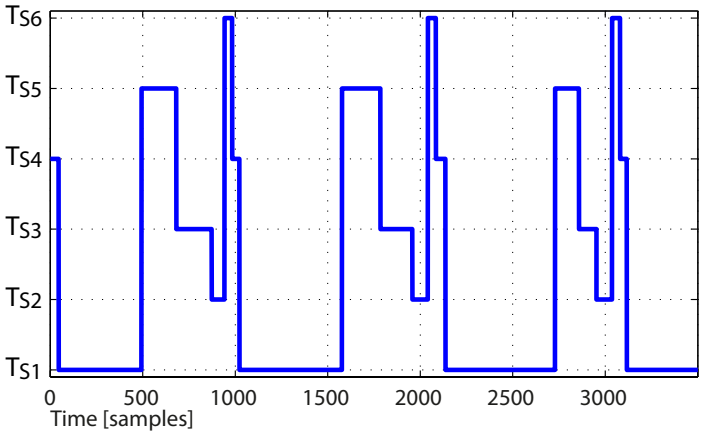


***Figure 6.14**: Sequence of matched clusters over time for an extract of the side view recordings in the discovery phase (also see the transition matrix in Figure 6.11 (b)).*

6.4.6 Comparison to Manual Annotation

A video extract of several workflow repetitions was viewed by an uninvolved person in order to describe the observed workflow in words. In Table 6.2 the annotated tasks are described and the reference is given for the two camera views. The automatically discovered tasks correspond very well to the tasks described by the human. Hence, the proposed technique is able to automatically extract tasks in a cyclic workflow, which are meaningful to human observers.

Manual task description	Side	Overhead
Two workers are putting a number of small spare parts (8 components) [...]	T_{S2}	T_{O1}, T_{O2}
Also they carry 2 big spare parts in the same table.	T_{S6}, T_{S4}	T_{O5}, T_{O3}
They are providing welding of the spare parts on the table construction.	T_{S1}	T_{O6}
One of them is manipulating and drives a yellow crane for taking the skeleton of the car in another plant.	T_{S5}	T_{O4}
This is the end of the workflow. The table plant is empty again and the workers start again [...].	T_{S3}	T_{O7}

***Table 6.2**: Comparison of our automatically detected workflow tasks with manual annotations.*

6.5 Workflow Interpretation

Using the model established in the previous section, the analysis of new videos can provide statistical information on the tasks carried out, or it enables the detection of abnormalities in the observations.

6.5.1 Modeling the Workflow

Task classification. Our workflow discovery technique provides task labels to the initially unlabeled image sequence. With this information, any supervised classification method can be trained. We show here a very simple implementation.

Training. Task classification is an multi-class classification problem. After PCA preprocessing, we opt to learn a representation of the labelled training data using Linear Discriminant Analysis (LDA) [Bishop 2007]. The LDA subspace, shown in Figure 6.15, is discriminative and arranges the data in compact clusters $\mathcal{C}$.

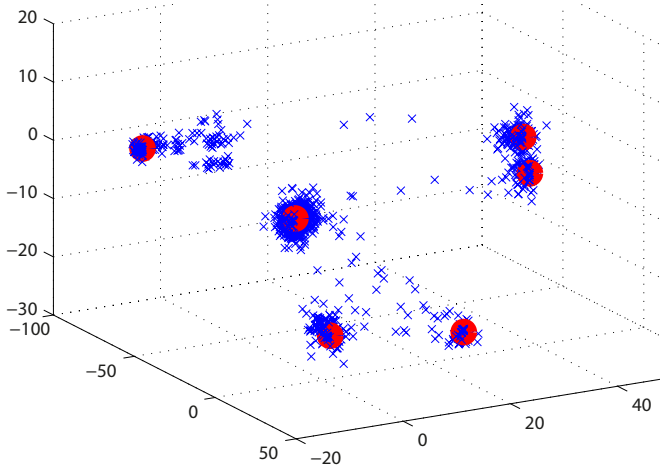


Figure 6.15*: The six established tasks form compact clusters in the LDA space. At runtime, images are analyzed in this space*

Runtime. At runtime, an image $\boldsymbol{x}$ is first projected into the LDA space to $\boldsymbol{x}'$. Then, the closest cluster center $\boldsymbol{c} \in \mathcal{C}$ determines the task label, *i.e.*,

$$T^*(\boldsymbol{x}') = \arg\min_{\boldsymbol{c}\in\mathcal{C}} ||\boldsymbol{x}' - \boldsymbol{c}||_2. \tag{6.3}$$

6.5.2 Anomaly Types

Three types of abnormalities can be detected with this simple model:

Appearance: Images which cannot be well assigned to any of the established clusters are considered as abnormal. This might be due to camera failures, large movements of the cameras or abnormal incidents in scene. To this end, we use the reconstruction error of the PCA model from the preprocessing step.

Sequence: The learned task sequence in the workflow should also be respected at runtime. If the task order changes, or a task is skipped, a problem can be signaled.

Timing: Each task is carried out for a certain duration. If the observed duration differs significantly from the trained one, a manufacturing issue might have occurred in this task.

We emphasize that a more sophisticated model of the workflow could certainly be learned. Typically, techniques which rely on Hidden Markov Models are appropriate. This is beyond the scope of this section, as the principal aim was to use our techniques for automatic workflow discovery.

6.5.3 Runtime Processing

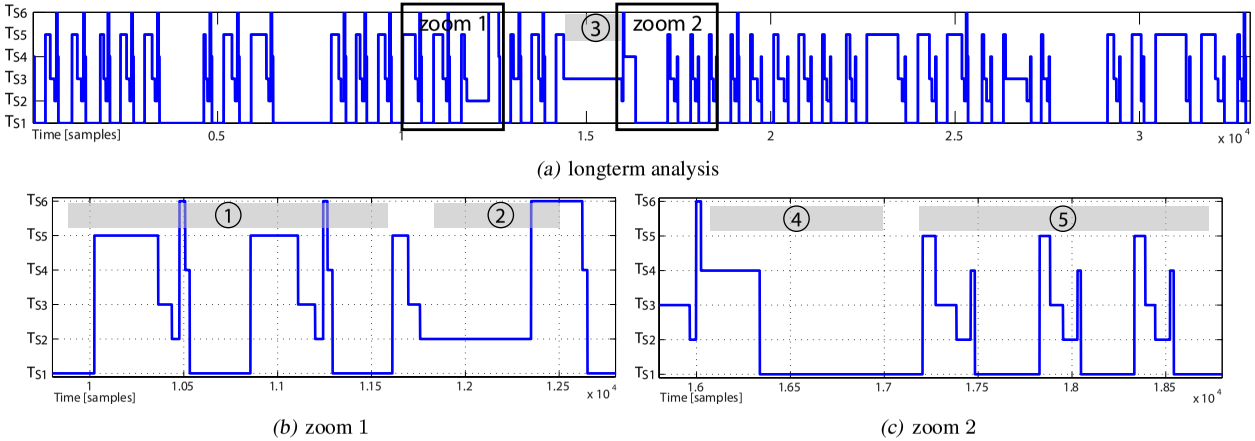
For the side view camera, we apply the established workflow model for runtime interpretation of the recordings obtained from a full working day. We process approximately $40,000$ frames and present the re-detected tasks plotted over time in Figure 6.16 (a). Figure 6.16 (b) and (c) show zooms of the long sequence, such that details become visible.

Statistics. In regular working cycles, the tasks are executed at regular speeds. Hence it is interesting to estimate the duration of each task from the data. A box-plot of timings for each task is shown in Figure 6.17. T_{S4} and T_{S6} for example, are short and very regular. They correspond to the placement of the first and second long metallic bar, respectively. In the long task T_{S1} all the metallic parts are welded together. T_{S5} has a very variable timing. In this task, the assembled parts are delivered with a crane to another plant. Since it depends on the advancement of neighboring working cells, pauses occur in this task and its timing seems unpredictable.

Processing time. Since we only perform linear operations on the input features to project them into subspaces, the proposed analysis technique is very efficient. The actual algorithm runs at more than 25 fps on a standard PC using our MATLAB implementation.

6.5.4 Detected Anomalies

During the runtime processing, several interesting cases are detected automatically:



(a) longterm analysis

(b) zoom 1

(c) zoom 2

***Figure 6.16**: Analysis of unseen data with the learned workflow model. The matched tasks are plotted over time. One working day of video is analyzed and different anomalies are spotted. The markers refer to the descriptions in the text and Figure 6.18 and Figure 6.19.*

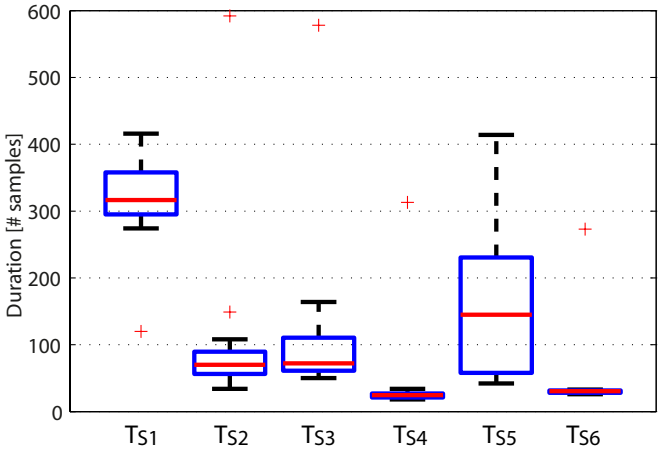


***Figure 6.17**: Box-plot for the duration of the tasks in the workflow. Task visit timings that are statistical outliers are indicated with crosses. Please note that for better visibility, the y-axis is bounded, and not all outliers (crosses) are shown.*

Appearance. Figure 6.18 (a) shows an exemplary abnormal event, which is detected because the image appearance does not apply well to any cluster. In this image, welding sparks appear at a very abnormal location, and we can suspect something abnormal going on here.

Timing. From the duration statistics in Figure 6.17, abnormal timings of tasks can be identified. Three such cases are shown in Figure 6.18 (b), (c) and (d). They correspond to the markers ②, ③ and ④ in the plots in Figure 6.16, respectively and shows a work break at an unnatural instant within the working cycle, a worker shift change and a break for cleaning of the production space.

Sequence. During the analyzed work day, the sequential pattern of executed tasks changes. This appears from the comparison of markers ① and ⑤ in Figure 6.16. A closer look is provided in Figure 6.19, where it can be seen that two tasks are interchanged. During the workflow discovery process in the morning, the long metallic bar was first placed on the left, then on the right. Later on however, this learned cyclic structure of the workflow is no longer respected. The modification occurs right after the change of shift (marker ③). Apparently,

(a) Sparks at an abnormal location

(b) Coffee break, ②

(c) Shift change, ③

(d) Cleaning, ④

***Figure 6.18**: Abnormal appearance (a) and abnormal timing (b)-(d) is detected automatically in the analyzed video.*

the workers in the afternoon shift prefer to invert the order of the placement. This is not a critical issue here, but it could have been one. Nevertheless If the workflow discovery algorithm is run for a longer period of time, our approach will respect this task switch and will merge those two clusters. This would also be in line with the human interpretation of Table 6.2.

6.5.5 Discussion

As has been shown, our proposed algorithm is able to automatically extract meaningful workflows. But what information in the images is really used? Since SFA is used in the discovery process, the structure of the embedding provides some information. The first two Eigenimages obtained by SFA projection are depicted in Figure 6.20. As can be seen, the variance encodes the motion on the assembly table, which can be interpreted as the presence or ab-

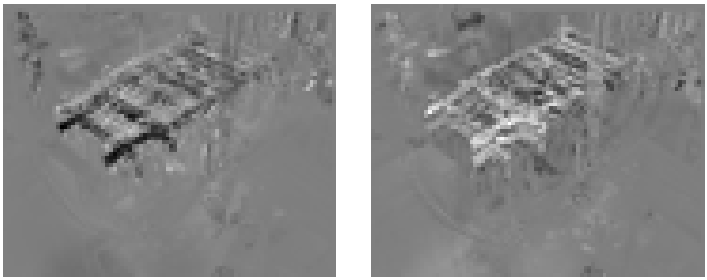
(a) morning shift ① *(b)* afternoon shift ⑤

***Figure 6.19**: Inverted tasks for morning (a) and afternoon shift (b). The placement order of the two long metallic bars is switched.*

sence of the parts. In contrast to many other methods which detect and track people, our approach does not focus on humans. The workers might even be considered as noise with respect to the entire workflow. Admittedly, they are somehow implicitly modeled, since they are necessary to bring the parts along.

We point out that this is not a general claim, but surly depends on the actual scenario. For another working cell, a person or other objects would well define the workflow. Due to the general structure of our data-driven algorithm, they

would be picked up automatically in such cases. In summary, our algorithm chooses to model the workflow in the easiest possible way and respects the assumptions.



(a) first Eigenimage *(b)* second Eigenimage

***Figure 6.20**: Eigenimages of SFA (gray values corresponds to zeros, black to negative values and white to positive values). High variance is found on the assembly table, which can be interpreted as the presence or absence of parts.*

6.6 Conclusions

In this chapter, we have demonstrated the validity of our techniques for diverse surveillance applications. We have exploited and adapted the approach of Chapter 5 in order to interpret traffic scenes, webcam videos and industrial workflows. The arisen models of normal scene behavior turn out to be semantically interpretable and accurate for the detection of normal and abnormal situations in previously unseen data. As the developed approach is generic and relies on the temporal structure of the underlying activities, the use of specifically tuned features is no longer required. Indeed, we have successfully used different image descriptors.

As we have shown in Section 6.2 and 6.3, the underlying technique is very generic and applicable for different types of surveillance footage. One step further, in Section 6.4, we make use of additional constraints for model building. In the presented case, these constraints are induced by the repeated cyclic patterns in the workflow. We explicitly enforce them, while still respecting temporal consistency and demonstrate that without human interaction nor parameter tuning, cyclic workflows can be extracted robustly

Our unsupervised models of surveillance scenes capture the behavior that regularly occurs in the monitored scene. With simple generative modeling techniques, we can then interpret new videos and detect unseen behavior configurations as abnormal events. Successful cases include for example the discovery of abnormal ambulance traffic (Figure 6.3), a car irregularly driving on a zebra crossing (Figure 6.5) or the camera failure and the street festival (Figure 6.7). A number of abnormal situations in the workflow have also been reported, there additionally taking into account the duration and the ordering of the workflow tasks. However, as the scene is modeled in a holistic manner, only events that are significant enough will be detected. For example, a pedestrian irregularly crossing the QMU Junction (Section 6.2.1, *c.f.* [Hospedales *et al.* 2009]), was not reported. However, real abnormal events often influence the scene in a larger area. If an accident happened in the case of the pedestrian crossing the street, this would certainly have been noticed, as many cars would have been forced to adapt and modify their driving.

7

Conclusions

In this thesis, we were interested in supervised and unsupervised techniques for the analysis of surveillance videos and the detection of abnormal behavior in these videos. In particular, we wanted to interpret human motion in indoor scenes. To this end, we have presented three independent approaches in Chapters 2, 4 and 5. Chapter 3 was an extension to the tracker-tree of Chapter 2 and Chapter 6 presented applications using the technique of Chapter 5 for different surveillance scenarios that went beyond human motion analysis. This has proven the wide applicability of the developed approach. Here we summarize the techniques, present a few key-insights and finally sketch the possible perspectives.

7.1 Summary and Comparison

Table 7.1 summarizes and compares the presented approaches with respect to some key-properties. In the following, each chapter is reviewed briefly.

Tracker-Trees (*Chapter 2*). We have developed tracker-trees as a way to reason among more general and more specific visual trackers. Each tracker in the tree was trained such that it incorporates clear assumptions about what it expects to observe. The goal was not to improve tracking robustness, but to make a reasoning from the configuration of more or less confident trackers. This enabled anomaly detection in a principled manner. We have shown this in experiments, which target the indoor behavior interpretation of (elderly) persons. The tracker-tree was useful, not only for abnormal event detection, but also for the re-detection of familiar activities. The few observed failure cases were

	Tracker-Trees *Chapter 2*	T-T Update via T-L *Chapter 3*	Cascaded Hierarchies *Chapter 4*	Temporal Relations *Chapter 5*
Training	Fully supervised	Few manual annotation	Unsupervised	Unsupervised
Hierarchy type	Fix, designed for the application	Expandable to some extent	Data-driven, but fixed structure	Data-driven, data-dependent structure
Model	Subspace trackers & state-of-the-art techniques		Clustering in high-dim. space	Subspace (SFA & PCA)
Abnormal event detection	Tracker confidence scores		data-dependent outlier	PCA reconstruction error
Update during runtime	No	Yes, manual interaction	Yes, with hierarchical extension	Not shown, but feasible as in Chapter 4
Application scenarios	Human motion	Human motion	Human motion, other scenarios possible	Human motion, traffic webcams and workflows.

***Table 7.1**: Tabular summary of the presented methods for abnormal behavior detection in hierarchies.*

mainly due to erroneous background subtraction. To cope with challenging real-life issues such as changing illumination, cast shadows or moving furniture, the system would certainly need to be tuned and augmented with more suitable and robust trackers. However, we have demonstrated that the tracker-tree concept shows good performances in our setting and could certainly also be applied for other surveillance scenarios, if suitable trackers are available.

Tracker-Tree Update via Transfer Learning (*Chapter 3*). In order to facilitate the training procedure of the activity trackers used in the tracker-tree, we successfully integrated a transfer-learning framework. Using prior knowledge from the known activities together with few human annotations (usually one per class), we have shown that the method accurately labels new activity concepts. This is especially useful if true abnormal events need to be re-detected later and require an accurate semantic label.

Cascaded Hierarchies for Appearance and Motion (*Chapter 4*). We established two hierarchies for the modeling of human behavior in an unsupervised manner. The first hierarchy uses top-down clustering to refine the description of appearances, while the second hierarchy builds on the output of the first one to encode motion components in a bottom-up manner. In fact, we have shown that such a two-stage model is bio-inspired, which was also experimentally verified in a behavioral study conducted with monkeys. Even though clear parallels were observed, monkeys have a better capacity to integrate the training in a broad manner, whereas a computational model tends to over-fit. For surveillance tasks, experiments conducted on indoor activity videos have shown the validity of our hierarchical model, both in terms of abnormal event detection performance and interpretation. We have additionally demonstrated that an update during runtime is feasible and useful.

Temporal Relations in Activities (*Chapter 5*). We have observed that the underlying temporal structure of activities provides a strong cue in (human) activity analysis. The use of Slow Feature Analysis enables the incorporation of these temporal relations during the modeling process. We have demonstrated how to automatically create a hierarchical model, which also leads to an intrinsic definition of activities from the available training data. The discovered activities often even have a semantic meaning, that could be labeled with little effort. Applied to unseen data, such a temporally coherent model exhibits superior performances for abnormal event detection, compared to the previous technique.

Furthermore, in *Chapter 6* we have demonstrated that thanks to the very generic concepts, which were incorporated in the modeling process, the technique of Chapter 5 is also applicable in different surveillance settings and tasks. With little modification with respect to the features, we have shown street scene analysis and webcam monitoring with very low frame-rate. Finally, in an industrial scenario, we have extracted meaningful workflows, and used the models for abnormal behavior detection.

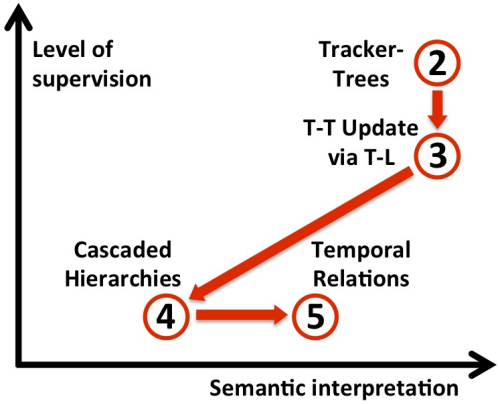


***Figure 7.1**: Qualitative summarization of the techniques developed in this thesis with respect to two key-aspects. The numbers refer to the chapters.*

Analyzing these summaries, we qualitatively recapitulate the developed techniques in Figure 7.1. In this two-dimensional plot, each technique is located with respect to the required level of supervision and its capacity for semantic interpretation of the analyzed behavior. The approaches of Chapter 2 and Chapter 3 require labeled training data and therefore incorporate semantic information. The approaches of Chapter 4 and Chapter 5 are unsupervised and have the advantage of learning autonomously from the available data. Including the temporal relations in the modeling process increases the level of semantic interpretability.

7.2 Insights

Besides many lessons learned specific to the outcomes and performances of the developed methods, we have identified a few valuable key-insights.

Behavior modeling in hierarchies. All the presented approaches for abnormal behavior detection in surveillance videos rely on a hierarchical modeling of the normally observed behavior. The benefit of the hierarchical encoding has been shown in different experiments, as it outperforms flat-structured models of similar or higher complexity. This is true equally for human activity modeling, the analysis of traffic scenes or the interpretation of webcam streams. The proposed hierarchical reasoning on anomalies is principled and robust, and furthermore paves the way for semantic interpretation. Abnormal events that are detected closer to the root tend to be more severe than those identified closer to the leaves. In that sense, we have also shown that the detection of abnormal events is possible if we follow the indirect route of modeling normality and detecting outliers to these models.

Use of temporal information. In particular in Chapter 5 and the applications in Chapter 6, we have built on a modeling technique that explicitly includes the temporal relations in the model building process. The observed activities are analyzed not only with respect to their appearance, but also the characteristics of consecutive differences are taken into account. In previous works as well as in the approach of Chapter 4, such cues are either encoded in the features or on a higher level analysis stage that connects (modeled) appearances. In contrast, if we take into account the temporal cues to build the hierarchy, this model reflects well the observed activity concepts, mostly in a semantically meaningful manner.

Data-driven models. The methods of Chapter 4 and 5 build a bottom-up representation of normal behavior in an unsupervised manner. Hence, the frequently observed data determines what is expected to happen in the future. We have found a way to make these data-driven models as accurate and meaningful as possible and have shown how far we can use them for anomaly detection. Obviously, such models lack of higher-level semantic and contextual understanding of the observed scene, which limits their applicability in some used-cases. For example, a well-defined, but hard-to-detect concept such as an abandoned piece of luggage would not be detected with such a bottom-up approach.

7.3 Perspectives

The questions answered and the methods developed in this thesis raise multiple new issues. Here we propose a few promising lines of future work, which could further improve the quality of abnormal behavior detection and shape the techniques even more towards practical applicability.

Scene context. Deduced from the last insight above, including contextual information is useful in many cases. One could learn for example which behavior happens in which part of then monitored living room. In the same way, the timing of activities can also be included. A detected person walking in the living-room at 2 a.m. would then be less usual compared to daytime motion. One step further, if the context of a scene is precisely known before training (for example living-room *vs.* bedroom), the behavior models could be learned accordingly and abnormal event detection would be more robust and semantically meaningful.

Relevance of the detected events. Some types of abnormality can only be inferred from semantical reasoning withing an activity scene. For example, when it comes to the correct or wrong usage of objects such as for example walking aids, higher level knowledge is required. Equally, some detected events might not be relevant since they are not really abnormal. One such example is the presence of pets in elderly people's houses, as their behavior and especially their interaction with the persons is very hard to model and interpret accurately. Here also, some more informed detector would help. A future task would therefore be to combine bottom-up data-driven models with top-down contextual and semantical information.

Features types. Since the goal of the presented work was to automatically detect abnormal behavior, we were not principally concerned with the representation of videos and images. For many applications, the employed feature types can certainly be improved, especially the silhouette features are prone to failure. In the case of indoor monitoring, one option is to additionally use data from depth cameras (*e.g.*, Microsoft Kinect) for improved robustness. Furthermore, the level of detail needs to be adapted with respect to the abnormal events that shall be detected. For instance, if abnormal pedestrian motion on street scenes is of interest, our coarse scene representation used in the webcam experiment (Section 6.3) is clearly insufficient.

Long-term adaptation. Admittedly, we have tested most of the presented approaches on comparably short video sequences. Only for the Times-Square webcam stream, the data recordings of several weeks was analyzed. Therefore it remains to be shown how our data-driven approaches perform in long-term. To this end however, they would need to incorporate principled update routines, that permit the models to adapt over time and incorporate very large amounts of data.

Bibliography

[Adam *et al.* 2008] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz. Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):555–560, 2008. 1.2

[Ali and Shah 2007] S. Ali and M. Shah. A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 1.1

[Anderson *et al.* 2009] D. Anderson, R. H. Luke, J. M. Keller, M. Skubic, M. Rantz, and M. Aud. Linguistic summarization of video for fall detection using voxel person and fuzzy logic. *Computer Vision and Image Understanding*, 113(1):80–89, 2009. 1.3

[Andriluka *et al.* 2008] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 1.1

[Basharat *et al.* 2008] A. Basharat, A. Gritai, and M. Shah. Learning object motion patterns for anomaly detection and improved object detection. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 1.1, 1.2

[Belongie *et al.* 2002] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002. 4.2.2

[Bishop 2007] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2007. 2.6.1, 5.3.1, 5.3.4, 6.4.3, 6.5.1

[Blum *et al.* 2010] T. Blum, H. Feussner, and N. Navab. Modeling and segmentation of surgical workflow from laparoscopic video. In *Proceedings Conference on Medical Image Computing and Computer Assisted Intervention*, 2010. 6.4.1

[Boiman and Irani 2005] O. Boiman and M. Irani. Detecting irregularities in images and in video. In *Proceedings International Conference on Computer Vision*, 2005. 1.2

[Bosch *et al.* 2007] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *Proceedings International Conference on Image and Video Retrieval*, 2007. 3.4.2

[Bradski 1998] G. R. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, (Q2), 1998. 2.4.1, 2.5.1, 4.3.2, 4.4.1

[Breitenstein *et al.* 2009a] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *Proceedings International Conference on Computer Vision*, 2009. 1.1

[Breitenstein *et al.* 2009b] M. D. Breitenstein, H. Grabner, and L. Van Gool. Hunting Nessie – Real-time abnormality detection from webcams. In *Proceedings IEEE Workshop on Visual Surveillance*, 2009. 1.2, 6.3, 6.3.1, 6.3.3

[Breitenstein 2009] M. Breitenstein. *Visual Surveillance: Dynamic Behavior Analysis at Multiple Levels*. PhD thesis, ETH Zürich, 2009. 1.1

[Chandola *et al.* 2009] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computer Surveys*, 41:15:1–15:58, 2009. 1.2

[Chen *et al.* 2005] J. Chen, K. Kwong, D. Chang, J. Luk, and R. Bajcsy. Wearable sensors for reliable fall detection. In *Proceedings IEEE Conference of the Engineering in Medicine and Biology Society*, 2005. 1.3

[Comaniciu and Meer 2002] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603 –619, 2002. 6.4.3

[Cook and Das 2007] D. J. Cook and S. K. Das. How smart are our environments? an updated look at the state of the art. *Pervasive and Mobile Computing*, 3(2):53 – 73, 2007. 1.3

[Cristianini and Shawe-Taylor 2000] N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines (and other kernel-based learning methods)*. Cambridge University Press, 2000. 3.3

[Cucchiara *et al.* 2003] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Trans-*

actions on Pattern Analysis and Machine Intelligence, 25(10):1337 – 1342, 2003. 1.1

[Cucchiara *et al.* 2005] R. Cucchiara, C. Grana, A. Prati, and R. Vezzani. Probabilistic posture classification for human-behavior analysis. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 35(1):42–54, 2005. 1.3

[De Luca *et al.* 2011] V. De Luca, H. Grabner, L. Petrusca, R. Salomir, G. Szekely, and C. Tanner. Keep breathing! Common motion helps multimodal mapping. In *Proceedings Conference on Medical Image Computing and Computer Assisted Intervention*, 2011. 6.4.3

[Dee and Velastin 2008] H. Dee and S. Velastin. How close are we to solving the problem of automated visual surveillance? *Machine Vision and Applications*, 19(5-6):329–343, 2008. 1.1

[Dollar *et al.* 2005] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Proceedings IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2005. 1.1

[Doucet *et al.* 2000] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte-carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10:197–208, 2000. 2.3.2

[Efros *et al.* 2003] A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Proceedings International Conference on Computer Vision*, 2003. 1.1

[Elgammal and Lee 2004] A. M. Elgammal and C. S. Lee. Inferring 3D body pose from silhouettes using activity manifold learning. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2004. 2.3

[Escobar *et al.* 2009] M.-J. Escobar, G. Masson, T. Vieville, and P. Kornprobst. Action recognition using a bio-inspired feedforward spiking network. *International Journal of Computer Vision*, 2009. 4.1

[Felzenszwalb *et al.* 2008] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 2.4.1, 2.5.1

[Fidler and Leonardis 2007] S. Fidler and A. Leonardis. Towards scalable representations of object categories: Learning a hierarchy of parts. In *Proceed-*

ings IEEE Conference on Computer Vision and Pattern Recognition, 2007. 4.2.3

[Fidler *et al.* 2006] S. Fidler, G. Berginc, and A. Leonardis. Hierarchical statistical learning of generic parts of object structure. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2006. 4.2.3

[Frost & Sullivan 2008] Frost & Sullivan. Video analytics - more intelligent surveillance. `www.frost.com`, Aug 2008. 1.1

[Gammeter *et al.* 2008] S. Gammeter, A. Ess, T. Jaggli, K. Schindler, B. Leibe, and L. Van Gool. Articulated Multi-body Tracking under Egomotion. In *Proceedings European Conference on Computer Vision*, 2008. 2.3

[Gibbons 1985] J. Gibbons. *Nonparametric statistical inference*. New York: Marcel Dekker, 1985. 3.4.2

[Giese and Poggio 2003] M. A. Giese and T. Poggio. Neural mechanisms for the recognition of biological movements. *Nature Reviews, Neuroscience*, 4:179–192, 2003. 4.1

[Gong *et al.* 2011] S. Gong, C. Loy, and T. Xiang. Security and surveillance. In *Visual Analysis of Humans: Looking at People*, pages 455–472. Springer, 2011. 1.1

[Gorelick *et al.* 2007] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247 –2253, 2007. 1.1

[Havlena *et al.* 2009] M. Havlena, A. Ess, W. Moreau, A. Torii, M. Jancosek, T. Pajdla, and L. Van Gool. Awear 2.0 system: Omni-directional audio-visual data acquisition and processing. In *Proceedings IEEE Workshop on Egocentric Vision*, 2009. 2.5.5, 3.4.1

[Hendel *et al.* 2010] A. Hendel, D. Weinshall, and S. Peleg. Identifying surprising events in video using bayesian topic models. In *Proceedings Asian Conference on Computer Vision*, 2010. 1.2, 6.2.2

[Hospedales *et al.* 2009] T. Hospedales, S. Gong, and T. Xiang. A markov clustering topic model for mining behaviour in video. In *Proceedings International Conference on Computer Vision*, 2009. 1.1, 1.2, 5.1, 6.2.1, 6.6

[Hu *et al.* 2004] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on*

Systems, Man and Cybernetics, Part C: Applications and Reviews, 34:334–352, 2004. 1.1

[Hu *et al.* 2006] W. Hu, X. Xiao, Z. Fu, D. Xie, F.-T. Tan, and S. Maybank. A system for learning statistical motion patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1450–1464, 2006. 1.1, 1.2

[Hu *et al.* 2011] D. H. Hu, V. W. Zheng, and Q. Yang. Cross-domain activity recognition via transfer learning. *Pervasive Mobile Computing*, 7:344–358, 2011. 3.2

[Huang *et al.* 2008] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *Proceedings European Conference on Computer Vision*, 2008. 1.1

[Jaeggli *et al.* 2007] T. Jaeggli, E. Koller-Meier, and L. Van Gool. Learning generative models for multi-activity body pose estimation. In *Proceedings Asian Conference on Computer Vision*, 2007. 2.3

[Jain *et al.* 1999] A. Jain, M. Murty, and P. Flynn. Data clustering: A review. *ACM Computer Surveys*, 31(3):264–323, 1999. 4.2.1

[Jhuang *et al.* 2007] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *Proceedings International Conference on Computer Vision*, 2007. 4.1

[Jie *et al.* 2011] L. Jie, T. Tommasi, and B. Caputo. Multiclass transfer learning from unconstrained priors. In *Proceedings International Conference on Computer Vision*, 2011. 3.2

[Johnson and Hogg 1995] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. In *Proceedings British Machine Vision Conference*, 1995. 1.2

[Kim and Grauman 2009] J. Kim and K. Grauman. Observe locally, infer globally: A space-time mrf for detecting abnormal activities with incremental updates. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 1.2

[Klampfl and Maass 2009] S. Klampfl and W. Maass. Replacing supervised classification learning by Slow Feature Analysis in spiking neural networks. In *Neural Information Processing Systems*, 2009. 5.3, 5.3.1, 5.3.2

[Knorr and Ng 1998] E. M. Knorr and R. T. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proceedings International Conference on Very Large Data Bases*, 1998. 4.3.1

[Kratz and Nishino 2009] L. Kratz and K. Nishino. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 1.2

[Kuettel *et al.* 2010] D. Kuettel, M. D. Breitenstein, L. Van Gool, and V. Ferrari. Whats going on? Discovering spatio-temporal dependencies in dynamic scenes. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 1.1, 1.2, 5.1, 6.2.1

[Lampert *et al.* 2009] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 3.2

[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. 4.1, 4.5, 5.3.4

[Laptev 2005] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64:107–123, 2005. 1.1

[Lavee *et al.* 2009] G. Lavee, E. Rivlin, and M. Rudzsky. Understanding video events: a survey of methods for automatic interpretation of semantic occurrences in video. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 39, 2009. 1.1

[Lawrence 2003] N. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In *Neural Information Processing Systems*, 2003. 2.3.1

[Lee and Elgammal 2007] C.-S. Lee and A. Elgammal. Modeling view and posture manifolds for tracking. In *Proceedings International Conference on Computer Vision*, 2007. 2.3

[Li *et al.* 2007] R. Li, T.-P. Tian, and S. Sclaroff. Simultaneous learning of nonlinear manifold and dynamical models for high-dimensional time series. In *Proceedings International Conference on Computer Vision*, 2007. 2.3

[Li *et al.* 2009] Q. Li, J. Stankovic, M. Hanson, A. Barth, J. Lach, and G. Zhou. Accurate, fast fall detection using gyroscopes and accelerometer-

derived posture information. In *Proceedings International Workshop on Wearable and Implantable Body Sensor Networks*, 2009. 1.3

[Lin *et al.* 2009] Z. Lin, Z. Jiang, and L. S. Davis. Recognizing actions by shape-motion prototype trees. In *Proceedings International Conference on Computer Vision*, 2009. 4.1, 4.2.4, 4.5

[Liu *et al.* 2011] J. Liu, M. Shah, B. Kuipers, and S. Savarese. Cross-view action recognition via view knowledge transfer. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 3.2

[Makris and Ellis 2005] D. Makris and T. Ellis. Learning semantic scene models from observing activity in visual surveillance. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 35(3):397–408, 2005. 1.1, 1.2

[Markets & Markets 2011] Markets & Markets. Global video surveillance market, applications and management services forecasts (2010-2015). `www.marketsandmarkets.com`, Jan 2011. 1

[McLachlan and Krishnan 1997] G. J. McLachlan and T. Krishnan. *The EM algorithm and extensions / Geoffrey J. McLachlan, Thriyambakam Krishnan*. Wiley, New York :, 1997. 2.6.1, 5.3.1

[Mehran *et al.* 2009] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 1.1

[Moeslund *et al.* 2006] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2):90–126, 2006. 2.3

[Nait-Charif and McKenna 2004] H. Nait-Charif and S. J. McKenna. Activity Summarisation and Fall Detection in a Supportive Home Environment. In *Proceedings International Conference on Pattern Recognition*, 2004. 1.3

[Nasution and Emmanuel 2007] A. Nasution and S. Emmanuel. Intelligent video surveillance for monitoring elderly in home environments. In *IEEE Workshop on Multimedia Signal Processing*, 2007. 1.3

[Natarajan and Nevatia 2008] P. Natarajan and R. Nevatia. Online, real-time tracking and recognition of human actions. In *Proceedings IEEE Workshop on Motion and Video Computing*, 2008. 4.2.3

[Nater *et al.* 2009] F. Nater, H. Grabner, T. Jaeggli, and L. Van Gool. Tracker trees for unusual event detection. In *Proceedings IEEE Workshop on Visual Surveillance*, 2009. 2.1

[Nater *et al.* 2010a] F. Nater, H. Grabner, and L. Van Gool. Exploiting simple hierarchies for unsupervised human behavior analysis. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 4.1

[Nater *et al.* 2010b] 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 *(both first authors contributed equally)*. In *DIRAC Workshop at ECML/PKDD*, 2010. 4.1, 4.5

[Nater *et al.* 2011a] F. Nater, H. Grabner, and L. Van Gool. Temporal relations in videos for unsupervised activity analysis. In *Proceedings British Machine Vision Conference*, 2011. 5.1, 6.1

[Nater *et al.* 2011b] F. Nater, H. Grabner, and L. Van Gool. Unsupervised workflow discovery in industrial environments. In *Proceedings IEEE Workshop on Visual Surveillance*, 2011. 6.1

[Nater *et al.* 2011c] F. Nater, T. Tommasi, H. Grabner, L. Van Gool, and B. Caputo. Transferring activities: Updating human behavior analysis *(both first authors contributed equally)*. In *Proceedings IEEE Workshop on Visual Surveillance*, 2011. 3.1

[Niebles *et al.* 2008] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 79(3):299–318, 2008. 1.1, 5.1

[Niebles *et al.* 2010] J. C. Niebles, C.-W. Chen, and Li Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *Proceedings European Conference on Computer Vision*, 2010. 5.2

[Noury *et al.* 2007] N. Noury, A. Fleury, P. Rumeau, A. K. Bourke, G. O. Laighin, V. Rialle, and J. E. Lundy. Fall detection - principles and methods. In *Proceedings IEEE Conference of the Engineering in Medicine and Biology Society*, 2007. 1.3

[Padoy *et al.* 2009] N. Padoy, D. Mateus, D. Weinland, M.-O. Berger, and N. Navab. Workflow monitoring based on 3D motion features. In *Proceedings ICCV Workshop on Video-oriented Object and Event Classification*, 2009. 6.4.1

[Pellegrini *et al.* 2009] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Proceedings International Conference on Computer Vision*, 2009. 1.1

[Rashidi and Cook 2010] P. Rashidi and D. J. Cook. Mining and monitoring patterns of daily routines for assisted living in real world settings. In *Proceedings of International Health Informatics Conference*, 2010. 1.3

[Rashidi and Cook 2011] P. Rashidi and D. J. Cook. Activity knowledge transfer in smart environments. *Pervasive and Mobile Computing*, 7(3):331 – 343, 2011. 3.2

[Rasmussen and Williams 2006] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. The MIT Press, 2006. 2.3.1

[Rougier *et al.* 2006] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau. Monocular 3d head tracking to detect falls of elderly people. In *Proceedings IEEE Conference of the Engineering in Medicine and Biology Society*, 2006. 1.3

[Rougier *et al.* 2007] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau. Fall detection from human shape and motion history using video surveillance. In *Proceedings Advanced Information Networking and Applications Workshops*, 2007. 1.3

[Satkin and Hebert 2010] S. Satkin and M. Hebert. Modeling the temporal extent of actions. In *Proceedings European Conference on Computer Vision*, 2010. 4.2.3

[Schindler and Van Gool 2008] K. Schindler and L. Van Gool. Action snippets: How many frames does human action recognition require? In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 1.1, 4.1

[Stalder *et al.* 2010] S. Stalder, H. Grabner, and L. Van Gool. Cascaded confidence filtering for improved tracking-by-detection. In *Proceedings European Conference on Computer Vision*, 2010. 1.1

[Stauffer and Grimson 1999] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 1999. 1.1

[Stauffer and Grimson 2000] C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000. 1.1, 1.2, 5.1

[Stenger *et al.* 2009] B. Stenger, T. Woodley, and R. Cipolla. Learning to track with multiple observers. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 2.1

[Suykens *et al.* 2002] J. Suykens, T. V. Gestel, J. D. Brabanter, B. D. Moor, and J. Vanderwalle. *Least squares support vector machines*. World Scientific, 2002. 3.3

[Tenenbaum *et al.* 2000] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000. 2.3.1

[Tinetti 2003] M. E. Tinetti. Preventing falls in elderly persons. *New England Journal of Medicine*, 348(1):42–49, 2003. 1.3

[Tommasi *et al.* 2010] T. Tommasi, F. Orabona, and B. Caputo. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 3.3, 3.3, 3.3, 3.4.2

[Toyama and Hager 1999] K. Toyama and G. D. Hager. Incremental focus of attention for robust vision-based tracking. *International Journal of Computer Vision*, 35:45–63, 1999. 2.1

[Turaga *et al.* 2009] P. Turaga, A. Veeraraghavan, and R. Chellappa. Unsupervised view and rate invariant clustering of video sequences. *Computer Vision and Image Understanding*, 113(3):353–371, 2009. 1.1, 5.1, 5.3.3, 5.3.3

[Turek *et al.* 2010] M. W. Turek, A. Hoogs, and R. Collins. Unsupervised learning of functional categories in video scenes. In *Proceedings European Conference on Computer Vision*, 2010. 1.1

[Urtasun *et al.* 2006a] R. Urtasun, D. J. Fleet, and P. Fua. 3D people tracking with gaussian process dynamical models. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2006. 2.3

[Urtasun *et al.* 2006b] R. Urtasun, D. J. Fleet, and P. Fua. Temporal motion models for monocular and multiview 3D human body tracking. *Computer Vision and Image Understanding*, 104(2):157–177, 2006. 5.3.4

[van Kasteren *et al.* 2010] T. van Kasteren, G. Englebienne, and B. J. A. Kröse. Transferring knowledge of activity recognition across sensor networks. In *Pervasive*, pages 283–300, 2010. 3.2

[Vangeneugden *et al.* 2010] J. Vangeneugden, K. Vancleef, T. Jaeggli, L. Van Gool, and R. Vogels. Discrimination of locomotion direction in impoverished displays of walkers by macaque monkeys. *Journal of Vision*, 10(4):22.1–19, 2010. 4.5, 4.5.2, 4.5.3

[Veres *et al.* 2010] G. Veres, H. Grabner, L. Middleton, and L. Van Gool. Automatic workflow monitoring in industrial e nvironments. In *Proceedings Asian Conference on Computer Vision*, 2010. 6.2.1, 6.4.1

[Wang *et al.* 2006] X. Wang, K. Tieu, and E. Grimson. Learning semantic scene models by trajectory analysis. In *Proceedings European Conference on Computer Vision*. 2006. 1.1

[Wang *et al.* 2009] X. Wang, X. Ma, and W. E. L. Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):539–555, 2009. 1.1, 5.1

[Weinshall *et al.* 2012] D. Weinshall, A. Zweig, H. Hermansky, S. Kombrink, F. W. Ohl, J. Anemueller, J.-H. Bach, L. Van Gool, F. Nater, T. Pajdla, M. Havlena, and M. Pavel. Beyond novelty detection: Incongruent events, when general and specific classifiers disagree. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99(PrePrints), 2012. 1.4, 2.2, 2.2, 4.3.3, 5.4.2

[Wiskott and Sejnowski 2002] L. Wiskott and T. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, 14(4):715–770, 2002. 5.3, 5.3.2, 6.4.3

[Wiskott 2003] L. Wiskott. Slow feature analysis: A theoretical analysis of optimal free r esponses. *Neural Computation*, 15(9):2147–2177, 2003. 5.3.1, 5.3.1

[Wu and Nevatia 2007] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75:247–266, 2007. 1.1

[Xian-Ming and Shao-Zi 2009] L. Xian-Ming and L. Shao-Zi. Transfer AdaBoost learning for action recognition. In *IEEE International Symposium on IT in Medicine & Education*, 2009. 3.2

[Yang *et al.* 2010] W. Yang, Y. Wang, and G. Mori. *Learning Transferable Distance Functions for Human Action Recognition*, pages 349–370. Advances in Pattern Recognition. Springer, 2010. 3.2

[Yao *et al.* 2010] A. Yao, J. Gall, and L. Van Gool. A hough transform-based voting framework for action recognition. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 1.1

[Zhao and Nevatia 2004] T. Zhao and R. Nevatia. Tracking multiple humans in complex situations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1208 –1221, 2004. 1.1

[Zhong *et al.* 2004] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2004. 1.2

[Zhou *et al.* 2008] F. Zhou, F. De la Torre, and J. K. Hodgins. Aligned cluster analysis for temporal segmentation of human motion. In *Proceedings IEEE Conference on automatic face and gesture recognition*, 2008. 1.1

[Zhou *et al.* 2010] F. Zhou, F. De la Torre, and J. F. Cohn. Unsupervised discovery of facial events. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 5.1

[Zivkovic and van der Heijden 2006] Z. Zivkovic and F. van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27:773–780, 2006. 2.3

[Zouba *et al.* 2009] N. Zouba, F. Bremond, and M. Thonnat. Multisensor fusion for monitoring elderly activities at home. In *Proceedings IEEE Conference on Advanced Video and Signal Based Surveillance*, 2009. 1.3

Curriculum Vitae

Name	Fabian Emanuel Nater
Date of birth	14^{th} June 1981
Citizenship	Swiss

Education

2008 – 2012	Doctor of Sciences ETH	
	Computer Vision Laboratory, ETH Zurich	(CH)
2001 – 2006	Master of Sciences EPFL in Electrical Engineering	
	Ecole Polytechnique Fédérale de Lausanne (EPFL)	(CH)
1994 – 2001	Matura Typus C (natural sciences)	
	Schweizerische Alpine Mittelschule, Davos	(CH)
1998 – 1999	Baccalauréat de Français	
	Lycée André Boulloche, Livry Gargan, Paris	(FR)

Experience

2008 – 2012	ETH Zurich, Computer Vision Laboratory	(CH)
	Research Assistant, EU research project DIRAC	
2007 – 2008	Baumer electric AG, Frauenfeld	(CH)
	Industrial vision sensor development	
2007	Phonak Communications AG, Murten	(CH)
	Digital audio processing and acoustics development	
2006	World Radiation Center, Davos	(CH)
	Civil service, electronics HW and SW develompent	
2006	Philips Research, Eindhoven	(NL)
	Psychoacoustics research, master thesis	
2005	Baumer electric AG, Heidelberg	(DE)
	Electronics HW and SW development	

Publications

[1] D. Weinshall, A. Zweig, H. Hermansky, S. Kombrink, F. W. Ohl, J. Anemüller, J.-H. Bach, L. Van Gool, F. Nater, T. Pajdla, M. Havlena and M. Pavel. Beyond Novelty Detection: Incongruent Events, when General and Specific Classifiers Disagree. *IEEE Transactons on Pattern Analysis and Machine Intelligence (to appear)*, 2012.

[2] F. Nater, H. Grabner and L. Van Gool. Unsupervised Workflow Discovery in Industrial Enviroments. In *Proceedings IEEE Workshop on Visual Surveillance*, 2011.

[3] F. Nater*, T. Tommasi*, H. Grabner, L. Van Gool and B. Caputo (* denotes equal contribution). Transferring Activities: Updating Human Behavior Analysis. In *Proceedings IEEE Workshop on Visual Surveillance (Oral Presentation)*, 2011.

[4] F. Nater, H. Grabner, and L. Van Gool. Temporal Relations in Videos for Unsupervised Activity Analysis. In *Proceedings British Machine Vision Conference (Oral Presentation)*, 2011.

[5] F. Nater*, J. Vangeneugden*, H. Grabner, L. Van Gool, and R. Vogels (* denotes equal contribution). Discrimination of locomotion direction at different speeds: A comparison between macaque monkeys and algorithms. In *Proceedings ECML/PKDD Workshop on Detection and Identification of Rare Audio-Visual Cues (Oral Presentation)*, 2010.

[6] F. Nater, H. Grabner, and L. Van Gool. Visual abnormal event detection for prolonged independent living. In *Proceedings IEEE Healthcom Workshop on mHealth (Oral Presentation)*, 2010.

[7] F. Nater, H. Grabner, and L. Van Gool. Exploiting Simple Hierarchies for Unsupervised Human Behavior analysis. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[8] F. Nater, H. Grabner, T. Jaeggli, and L. Van Gool. Tracker Trees: Hierarchies to spot rare events. In *Proceedings International Conference on Cognitive Systems (Oral Presentation)*, 2010.

[9] J. Breebaart, F. Nater, and A. Kohlrausch. Spectral and spatial parameter resolution requirements for parametric, filter-bank-based HRTF processing. In *Journal of the Audio Engineering Society*, 58(3):126-140, 2010.

[10] F. Nater, H. Grabner, T. Jaeggli, and L. Van Gool. Tracker Trees for Unusual Event Detection. In *Proceedings IEEE Workshop on Visual Surveillance (Oral Presentation)*, 2009.

[11] J. Breebaart, F. Nater, and A. Kohlrausch. Parametric binaural synthesis: Background, applications and standards. In Proceedings International Conference on Acoustics (NAG/DAGA), 2009.

[12] F. Marquis, B. Heldner, F. Nater, G. Biundo Lotito and R. Arnet. Method and system for providing hearing assistance to a user. Patent Application No. WO2008138365 (A1), 2008.