Fast human behavior analysis for scene understanding

Citation for published version (APA):
Lao, W. (2011). *Fast human behavior analysis for scene understanding*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Electrical Engineering]. Technische Universiteit Eindhoven. https://doi.org/10.6100/IR717716

DOI:
10.6100/IR717716

Document status and date:
Published: 01/01/2011

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Fast Human Behavior Analysis for Scene Understanding

Weilun Lao

Fast Human Behavior Analysis for Scene Understanding

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
Rector Magnificus, prof.dr.ir. C.J. van Duijn, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen op
12 oktober 2011 om 16.00 uur

door

Weilun Lao

geboren te Guangzhou, China

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr.ir. Peter H.N. de With

Copromotor:
dr. Jungong Han

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN
Lao, Weilun
Fast human behavior analysis for scene understanding / by Weilun Lao. –
Eindhoven : Technische Universiteit Eindhoven, 2011.
Proefschrift.

ISBN: 978-90-386-2790-8
NUR 959
Trefw.: digitale beeldverwerking / computer vision /
patroonclassificatie.
Subject headings: human motion / feature extraction / computer vision /
pattern classification.

Fast Human Behavior Analysis for Scene Understanding

Weilun Lao

Committee:

prof.dr.ir. P.H.N. de With	Eindhoven University of Technology, The Netherlands
dr. J. Han	Centre for Mathematics & Computer Science, The Netherlands
prof.dr.ir. A.W.M. Smeulders	University of Amsterdam, The Netherlands
prof.dr. J.J. Lukkien	Eindhoven University of Technology, The Netherlands
dr. L. Shao	University of Sheffield, United Kingdom
prof.dr. H. Corporaal	Eindhoven University of Technology, The Netherlands
prof.dr.ir. P.P.J. van den Bosch	Eindhoven University of Technology, The Netherlands

The work described in this thesis has been supported by two European R&D projects in the framework of the ITEA programme, called CANDELA and CANTATA.

Summary

Human behavior analysis has become an active topic of broad interest and relevance for a number of research and application areas. The research in recent years has been considerably driven by the growing level of criminal behavior in large urban areas and threat of terroristic actions. Also, accurate behavior studies have been applied to gaming, sports analysis systems and are emerging in healthcare.

When compared to conventional action recognition used in security applications, human behavior analysis techniques designed for embedded applications should satisfy the following technical requirements: (1) behavior analysis should provide scalable and robust results; (2) high-processing efficiency to achieve (near) real-time operation with low-cost hardware; (3) extensibility for multiple-camera setup including 3-D modeling to facilitate human behavior understanding and description in various events.

The key to our problem statement is that we intend to improve behavior analysis performance while preserving the efficiency of the designed techniques, to allow implementation in embedded environments. More specifically, we look into (1) fast multi-level algorithms incorporating specific domain knowledge, and (2) 3-D configuration techniques for overall enhanced performance. If possible, we explore the performance of the current behavior analysis techniques for improving accuracy and scalability. To fulfill the above technical requirements and address the research questions, we propose a flexible behavior analysis framework consisting of three processing layers: (1) pixel-based processing involving pixel-accurate background modeling, (2) object-based modeling for human detection, tracking and posture analysis, and (3) event-based analysis aiming at semantic event understanding. In Chapter 3, we specifically contribute to the analysis of individual human behavior. A novel body representation is proposed for posture classification based on a silhouette feature. Only pure binary-shape information is used for posture classification without texture/color or any explicit body models. To this end, we have studied an efficient HV-PCA shape-based descriptor with temporal modeling, which achieves a posture-recognition accuracy rate of about 86% and outperforms other existing proposals. As our behavior analysis scheme is efficient and achieves a

fast performance (6-8 frames/second), it enables a fast surveillance system or further analysis of human behavior. In addition, a body-part detection approach is presented. The color and body ratio are combined to provide clues for human body detection and classification, without using the conventional assumption of up-right body posture.

Afterwards, we design a specific framework for fast algorithms and apply this in two applications: tennis sports analysis and surveillance. Chapter 4 deals with tennis sports analysis and presents an automatic real-time system for multi-level analysis of tennis video sequences. First, we employ a 3-D camera model to create an accurate floor plane for semantics and tactics analysis. Second, a weighted linear model combining the visual cues in the real-world domain is proposed to identify various events. The experimentally found event extraction rate of the system is about 90%. Also, audio signals are combined to enhance the scene analysis performance. The complete proposed application is efficient enough to obtain a (near) real-time performance (2-3 frames/s for resolution of 720×576 pixels, and 5-7 frames/s for 320×240 pixels, with a P-IV PC running at 3 GHz).

Chapter 5 addresses surveillance and presents a full real-time behavior analysis framework, featuring layers at pixel, object, event and visualization level. More specifically, this framework captures the human motion, classifies its posture, infers the semantic event exploiting interaction modeling, and performs the 3-D scene reconstruction. The introduced system design is based on a specific software architecture, by employing the well-known "4+1" view model. In addition, human behavior analysis algorithms are directly designed for real-time operation and embedded in an experimental runtime AV content analysis architecture. This executable system is designed to be generic for multiple streaming applications with component-based architectures. To evaluate the performance, we have applied this networked system in a single-camera setup. The experimental platform operates with two Pentium Quadcore engines (2.33 GHz) and 4-GB memory. Performance evaluations have shown that this networked framework is efficient and achieves a fast performance (13-15 frames/s) for monocular video sequences. Moreover, a dual-camera setup is tested within the behavior-analysis framework. After automatic camera calibration is conducted, the 3-D reconstruction and communication among different cameras are achieved. The extra view in the multi-camera setup improves the human tracking and event detection in case of occlusion. This extension of multiple-view fusion improves the event-based semantic analysis by 8.3-16.7% in accuracy rate.

The detailed studies of two experimental intelligent applications, i.e., tennis sports analysis and experimental robbery detection for surveillance, have been tested and evaluated in the framework of the European Candela and Cantata ITEA research programs. Both systems demonstrated competitive performance with respect to accuracy and efficiency.

Samenvatting

De analyse van menselijk gedrag is in toenemende mate relevant voor diverse onderzoeksgebieden en toepassingen. Het onderzoek is gemotiveerd door de groei van crimineel gedrag in grote stedelijke gebieden en de dreiging van terroristische acties. Nauwkeurige gedragsstudies zijn tevens toegepast in systemen voor computerspel- en sportanalyse en het gebied is groeiend in de gezondheidszorg.

Vergeleken met conventionele actieherkenning in beveiligingssystemen, moeten technieken voor menselijke gedragsanalyse voor ingebedde toepassingen aan de volgende technische eisen voldoen: (1) de analyse moet schaalbaar en robuust zijn, (2) hoge efficiëntie is nodig voor (bijna) real-time verwerking, (3) uitbreidbaarheid voor multi-camera systemen inclusief 3-D modellering voor de analyse en beschrijving van diverse gebeurtenissen.

De kern van de probleemstelling is dat we de gedragsanalyse willen verbeteren met behoud van efficiëntie van de ontworpen technieken, zodat implementatie in ingebedde systemen is gegarandeerd. De specifieke onderzoeksaspecten zijn (1) snelle meervoudig gelaagde algoritmen gebruikmakend van specifieke domeinkennis, en (2) 3-D configuratietechnieken voor algemene verbetering van de analyse. Indien mogelijk evalueren we de prestaties van de huidige technieken voor gedragsanalyse m.b.t. nauwkeurigheid en schaalbaarheid. Rekening houdend met bovengenoemde technische eisen en de onderzoeksvragen, streven we naar een flexibel kader voor gedragsanalyse met drie verwerkingslagen: (1) pixel-gebaseerde verwerking in achtergrondsmodellering, (2) object-gebaseerde modellering voor detecteren, volgen en postuuranalyse van mensen, en (3) gebeurtenisanalyse voor interpretatie van de semantische betekenis. In hoofdstuk 3 leveren we een specifieke bijdrage aan de analyse van individueel menselijk gedrag. We presenteren een nieuw eenvoudig model van het menselijk lichaam voor postuurclassificatie, gebaseerd op het silhouet. Voor deze postuurclassificatie gebruiken we alleen binaire vorminformatie zonder textuur/kleur of enige expliciete lichaamsmodellen. Het algoritme gebruikt een efficiënte vormgebaseerde HV-PCA methode met tijdsmodellering, die met 86% nauwkeurigheid postuurherkenning uitvoert en bestaande voorstellen overtreft. Het gerealiseerde raamwerk is efficiënt en

bereikt hoge prestaties (6-8 beelden/s), zodat een real-time beveiligingssysteem of verdere gedragsanalyse mogelijk zijn. Tevens wordt een detectie van lichaamsdelen gepresenteerd, waarbij de kleur en lichaamsverhouding als aanwijzingen zijn gecombineerd voor menselijke lichaamsdetectie en classificatie, zonder de gebruikelijke aanname van een rechtopstaande lichaamshouding.

Hierna ontwerpen we een specifiek kader voor snelle algoritmen en passen dit toe in twee toepassingen: tennissportanalyse en beveiliging. Hoofdstuk 4 behandelt de analyse van tennissport en beschrijft een automatisch real-time systeem voor meerlaagse analyse van tennisvideo's. Ten eerste gebruiken we een 3-D cameramodel voor een modelgebaseerde analyse van spelgedrag en tactiek. Ten tweede beschrijven we een gewogen lineair model dat de visuele hints in het reële domein combineert om diverse gebeurtenissen te identificeren. Het experimenteel gevonden percentage van de gebeurtenisextractie van het systeem is ca. 90%. Tevens zijn geluidssignalen gebruikt om de prestaties van de scène-analyse te verbeteren. De complete toepassing is efficiënt genoeg voor (bijna) real-time verwerking (2-3 beelden/s met een resolutie van 720×576 pixels en 5-7 beelden/s met 320×240 pixels, met een P-IV PC op 3GHz).

Hoofdstuk 5 richt zich op beveiliging en presenteert een volledig real-time systeem voor gedragsanalyse met verwerking op pixel-, object-, gebeurtenis- en visualisatieniveau. Het systeem analyseert specifiek de menselijke beweging, classificeert het postuur en leidt hieruit de semantische betekenis af, gebruikmakend van interactiemodellering, en voert daarna de 3-D scènereconstrutie uit. Het ontwerp is gebaseerd op een specifieke SW architectuur volgens het bekende "4+1" model. Bovendien zijn de analyse-algoritmen direct ontworpen voor real-time toepassing en ingebed in een experimentele AV-analyse executiearchitectuur. De executie is generiek ontworpen voor verscheidene streamingtoepassingen met component-gebaseerde architecturen. Het systeem is geëvalueerd in een enkele-camera opstelling. Het experimentele platform gebruikt twee P-Quadcore processoren (2.33 GHz) en 4-GB geheugen. Experimenten hebben aangetoond dat dit systeem snel en efficiënt is (13-15 beelden/seconde) voor monoculaire videosequenties. Tevens is een dual-camera opstelling getest voor gedragsanalyse. Na de automatische camerakalibratie vindt de 3-D reconstructie en communicatie tussen de verschillende camera's plaats. Het extra gezichtspunt in de dual-camera opstelling verbetert het volgen van mensen en de gebeurtenisdetectie vooral bij occlusies. Deze uitbreiding en fusie van camerabeelden verhoogt de interpretatie van gebeurtenissen en semantische analyse met 8.3-16.7% in nauwkeurigheid.

De twee experimentele toepassingen voor de analyse van tennissport en experimentele overvaldetectie zijn gebruikt in verscheidene tests binnen de Europese ITEA Candela en Cantata onderzoeksprogramma's. Beide systemen hebben daarbij concurrerende prestaties laten zien met betrekking tot nauwkeurigheid en efficiëntie.

Contents

Chapter 1

Introduction

1.1 Background

Human behavior analysis has become an active topic of great interest and relevance to a number of applications and areas of research [36]. The research in recent years has been considerably driven by the growing level of criminal behavior in large urban areas and increase of terrorism actions. Besides this, accurate behavior studies have also found their ways in sports analysis systems. There are constant efforts in today's society to better understand how and why the human system responds to various stimuli, experiences and situations. Interpreting and understanding the way people move, react, and interact over time is the key to understanding and modeling the fundamentals of human nature, the kinematics and capabilities of the human body.

To analyze human behavior, one must first capture it and register it. For this task we may consider various input sources, which include visual, acoustic, or even pressure-based sensory systems. In this thesis, we focus on the analysis of moving video signals, particularly on the motion of individual persons and the motion-related interaction between persons. The research in this area has been conducted from various fields of interests, not only e.g. computer vision and computer graphics, but also the video surveillance and broadcasting industry. Computer vision researchers, on one hand, are interested in building object-based models and performing semantic analysis of real-world scenes captured by optical sensors. Computer graphics researchers, on the other hand, are looking forward to finding an attractive and cost-effective way of replicating the movements of human beings or deformable objects for computer-generated productions, such as games and movies.

Visual human-behavior analysis can be defined as the process of distinguishing people in a video scene, detecting and tracking those people over time, and conducting the semantic analysis of the scene and the people's behavior. The objective of the process is to understand and describe the human behavior based on video signals. The analysis can be extended to a three-dimensional representation of the motion activity for subsequent processing, with or without integrating actual scene knowledge. For example, for surveillance video, motion analysis has the objective to find answers to typical questions like: where are the people in the monitored scene and what are their activities?

In order to obtain the semantic interpretation of video content as described above, we first usually pay attention to primitive image signals (e.g. by analyzing pixel-based features). However, there is no straightforward approach for obtaining high-level semantics from the above pixel-based analysis results. In most cases, this gap is bridged by techniques addressed in an hierarchical level (i.e. pixel-based, object-based, event-based level). Although tremendous efforts have been spent in the past decades, a fast and automatic motion-analysis system for generic human behavior still does not exist. For specific applications, like professional sports analysis, models exist but they are proprietary and meet specific requirements. The objective of our research is that the analysis should be used in an embedded system and for a broad class of behaviors, such as home-use and security purposes.

With respect to object-based modeling, human bodies are typical examples of non-rigid objects, which means that their shapes and visibility of limbs have a large variety and typical change continuously over time. Motion-analysis techniques are typically used within pixel and object-based processing and provide inputs for semantic analysis tasks. Therefore, it is highly required to find a suitable model to represent object-level features, e.g. posture and shape. Then these object-based modeling lays a solid background for further semantic analysis.

In multi-person events, the semantic analysis is achieved by understanding the interactions among people involved in the scene. The events significantly rely on the temporal order and relationship of their position and/or posture changes. Therefore, it is necessary to impose appropriate spatial and temporal constraints and model them for each of the various two-person interaction patterns.

Furthermore, human behavior analysis plays a key role in various activities and the associated results can facilitate numerous applications. It is valuable to present validation model in various case studies. Given the large variety of applications, our aim is to study techniques for detection and classification of human behavior, which are validated in two different applications: sports analysis and surveillance.

In this thesis, we focus on the problem of behavior analysis and apply the techniques while integrating specific domain knowledge. We especially look

into performance issues of individual behavior analysis and propose an efficient scheme for semantic-level event analysis. It should be noted that this thesis is focused on (near) real-time implementation of consumer/embedded applications.

The remainder of this chapter is organized as follows. In Section 1.2, we briefly introduce the behavior analysis problem and present a layering of the required processing for human behavior analysis. We divide the behavior analysis into several layers, namely, pixel-based processing (background modeling), object-based modeling (human detection, tracking and posture analysis) and event-based analysis (event understanding). In Section 1.3, we discuss the performance requirements for designing behavior analysis systems, specifically for consumer/embedded use. Furthermore, we point out the performance deficiencies of current motion analysis systems and introduce our suggestions for improvements. In Section 1.4, we present the research objectives and specify our major contributions in behavior analysis. Section 1.5 provides the thesis structure and summarizes the chapters and their scientific background.

1.2 Automatic human behavior analysis

Overall, the growing interest in human behavior analysis is motivated by a wide spectrum of applications involving home multimedia, automatic surveillance, virtual reality, performance analysis, human computer interactions, and computer-generated animation. A summary of the possible applications is listed in Table 1.1.

Automatic surveillance provides the promise of detection, tracking and semantic analysis of multiple subjects with intelligent detection of activities of interest. Virtual reality applications meanwhile will be driven primarily by integrating more enriched forms of interaction with other participants or objects. They involve adding gestures, head pose and facial expressions as cues. Understanding human computer interactions is the key in developing next generation man-machine interfaces which are natural and intuitive to use. Performance analysis is extremely useful in the content-based video indexing and increasingly, in the field of movement analysis in sports and clinical studies. Motion-analysis techniques enable very low bit-rate video compression (e.g. MPEG-4) in object-based coding. Finally, computer generated animation, as we all know, is now a fast-growing and lucrative industry with its films depicting ever greater realism.

A typical behavior-analysis system usually consists of several processing steps, as depicted in Figure 1.1. These steps are: pixel-based processing, object-based modeling and event-based analysis. Each processing stage forms a layer in general behavior analysis and facilitates a wide range of applications.

Table 1.1: *Applications of behavior-analysis techniques.*

General domains	Specific areas
Automatic surveillance	Access control (Parking lots, supermarkets, department stores, etc.) Traffic control Security alarm (Banks, vending machines, ATMs, etc.)
Virtual reality	Interactive virtual worlds Games Virtual studios (e.g. advertising) Teleconferencing
Human computer interfaces	Social interfaces Sign-languages translation Gesture-driven control Signaling in highly noisy environments (airports, factories)
Performance analysis	Content-based indexing of sports video footage Personal training in golf, tennis, etc. Choreography of dance and ballet Clinical studies of human motion
Object-based coding	Very low bit-rate video compression
Computer-generated animation	3-D films production

1. Pixel-based processing

In the step of pixel-based processing, the background modeling is implemented. It is a crucial pre-processing step for visual behavior analysis from arbitrary video-capturing environments. A central question in pixel-based processing is how to optimally produce a background model from a dynamically changing background.

2. Object-based modeling

This stage performs human detection, trajectory estimation, body-based modeling (e.g. posture classification and skeleton-model generation). Each image within the video covering an individual human body is segmented to extract the 'blobs' representing foreground objects. These detected blobs are refined afterwards to produce the human silhouette. Human detection locates human areas from input video sequences. Every moving person in the scene is tracked over time. Afterwards, a body-based analysis is conducted to classify individ-

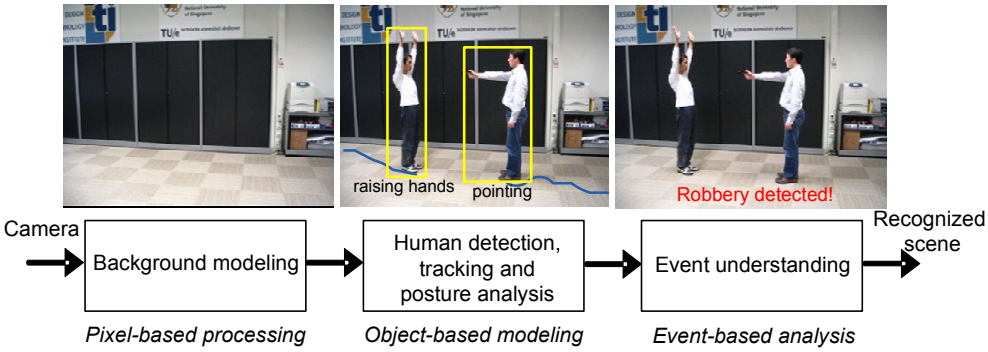

Figure 1.1: *Processing layers for a human behavior analysis system.*

ual activity. In some cases, it is necessary to obtain more detailed descriptions of these features, such as their appearances and contours.

3. Event-based analysis
Event-based analysis generates the final output of the complete behavior-analysis system: the semantic meaning of the event and possible indicators or parameters associated with this. In addition, interaction relationships are modeled to infer a multiple-person event. This semantic analysis is thus responsible for the event recognition. A key problem in event-based analysis is the broad variation of video sequences dealing with the same activity, as compared to that from different activities. Therefore, it is important to choose suitable classification technique that can distinguish the different events quite well.

Human behavior analysis has come a long way and the knowledge frontier of this domain has advanced tremendously. In some constrained cases, the techniques facilitate the design of a suitable application. It is, however, still very difficult to design a general and robust system which enables multiple real-world applications in real time. Approaches integrating different techniques are mostly designed for particular applications and are not flexible enough to be reused for other types of applications. Sub-domain problems such as developing accurate segmentation, performing robust human detection, handling occlusion situation and analyzing semantic-level behavior, are under continuous development.

In order to achieve a high-performance behavior-analysis system, each processing step in the system has to be carefully designed to satisfy specific application requirements. In the following section, we will detail such requirements.

1.3 Research requirements and problem statement

1.3.1 Research requirements

In this thesis, we focus on fast human behavior analysis for *embedded applications*. Motion analysis applied in consumer or surveillance applications offers many benefits such as wide applicability. Human behavior analysis techniques designed for embedded applications should satisfy the following technical requirements:

1. The behavior analysis should provide scalable and robust results for situations with a limited set of people.
Scalability is employed to offer the user various analysis results, ranging from a single alarm signal to a behavior description including movements, moving distances, etc. For this reason, it is desirable that the analysis system provides multiple-level results, which are extracted from the different processing layers, i.e. the pixel-level, object-level and event-level analysis and classification level. The processing levels or layers allow us to improve the processing quality in one layer without changing the total framework, so that we can increase the system robustness. We concentrate on the behavior of a limited set of people because high numbers of people are virtually impossible to process in real-time at present. More specifically, we have studied a surveillance application and tennis sports analysis as cases with a limited amount of active people in the scene.

2. High-processing efficiency achieving (near) real-time operation with low-cost hardware.
For consumer and embedded applications, (near) real-time performance is generally required while using low-cost hardware. Therefore, the algorithms have to be executed with limited processing resources and capacity. This means that algorithms have to be efficient and their complexity is constrained. Real-time execution is indispensable for surveillance applications, where the behavior sometimes should lead to direct action. In sports applications, the system should be fast enough to understand the proceeding of the game.

3. Extensibility for multiple-camera setup including 3-D modeling.
The use of multiple cameras is very helpful for a robust understanding of the human behavior, because the human body can be observed from multiple directions and occluding situations in one camera can be circumvented by using another camera view. It is beyond doubt that this will improve the success rate of the behavior interpretation. The use of multiple cameras allows to reconstruct the scene in 3-D so that the position of objects can be computed and a different view of actual events can be presented. More specifically, a

2D-3D mapping enables a computation of location and speed of objects and more detailed scene visualization. For example, these data can be used to create a top view of the scene where the motion and position of objects is shown for analysis purposes. This feature can significantly contribute to the scene understanding, like after-crime analysis and health-care behavior analysis of people. In this thesis, we intend to provide a scheme with integrated 2D-3D mapping for a surveillance application.

1.3.2 Problem statement

In this thesis, we will discuss a number of new techniques for each processing stage in experimental behavior analysis systems. To improve on the previously discussed requirements with respect to robustness, scalability and efficiency, we aim at the following research objectives:

1. *How can we efficiently represent the human body in order to facilitate real-time behavior analysis?*
2. *How should we efficiently use 3-D modeling for improved scene understanding?*
3. *How can we implement behavior analysis for a complete application and facilitate real-time execution?*

A brief summary of our problem statement is that we aim at improving the performance of the current behavior-analysis techniques by using models for the human body and the 3-D environment to facilitate efficient reasoning and improve the semantic understanding. We pay special attention to the efficiency of the designed techniques to allow implementation in embedded environments.

1.4 Research contributions

A. Individual behavior analysis
Chapter 3 addresses the problem of individual behavior analysis of a single human, involving action recognition and body-part detection. The contributions in this part are both on detecting body parts and efficient body representation.

1. *Effective body silhouette representation.* We propose a novel body representation based on a silhouette feature. Only pure binary-shape information is used for posture classification without texture/color or any explicit body models.

2. *Effective body-part detection.* To find the minimum amount of features for detecting human body, we present a scheme with only three features (body ratio, shape, color). This simple scheme is generic and integrated into a fast framework for body-part detection, without the conventional assumption of the human's posture being upright.

B. Tennis sports analysis application

In Chapter 4, we propose an AV-based tennis sports analysis system, featuring high-level scene analysis based on real-world visual or audio cues. The automatic sports analysis system can generate metadata that can be used for categorizing the sports video scenes and provide support for fast searching and retrieval of specific sports video. The contributions are both in creating a full application with human behavior analysis and in using audio signals to facilitate the detection of specific events that are difficult to find with video signals only.

- *Design of a complete three-layer framework for AV-analysis of tennis sports video.* We present one of the first fully automatic and real-time systems, which executes a joint combination of multi-level analysis of tennis video sequences and 3-D camera modeling. In addition, a weighted linear model combining the visual cues in the real-world domain is proposed to identify events. Adaptive adjustment of weight factors for each visual cue to different events ensures that our algorithm achieves a high accuracy.

- *Audio-based analysis of tennis sports video.* We extend the sports analysis framework by using audio signals for specific events. We propose a combination of a racket-hit detection and a parametric classifier driven by heuristic rules to classify services, returns and scores. Furthermore, we utilize a non-ball-tracking approach with the fusion of audio and video cues to infer the ball path. The tennis-ball path is generated for tennis-game tactics analysis without detection and tracking the ball itself.

C. Surveillance applications

In Chapter 5, we propose a flexible framework for event recognition for monocular video and multi-view video to study the influence of using more than one camera. Human interaction modeling and 3-D configuration are conducted for event understanding. The single camera or two-camera system is tested for different case studies for a surveillance application, leading to the following contributions.

- *Dual-camera surveillance system for behavior analysis with 3-D visualization and increased occlusion robustness.* The 3-D camera calibration

is modified to accommodate for employing two cameras. The 3-D information of objects, like location and speed, is used to reconstruct the scene for improved understanding, both in a simple top-view and in a more advanced way using posture models. We exploit the occurrence of occlusion and improve robustness by employing an alternative camera signal.

- *System validation for embedded real-time implementation.* To enforce and achieve processing efficiency, we have constructed an experimental real-time video-analysis system based on analyzing events with one or two cameras. This system was inserted in a component-based architecture and successfully executed for live demonstrations in the European ITEA project CANTATA. The whole system is executed on a single regular PC so that it is feasible for mapping into an embedded application.

1.5 Thesis outline and scientific background

Figure 1.2 briefly sketches the structure of the thesis. Besides the introductory and conclusion chapters (Chapter 1 and Chapter 6), the thesis consists of three parts in two different levels. At the level of fundamental techniques, overview of visual human motion analysis and individual person analysis are presented in Chapter 2 and Chapter 3, respectively. Afterwards, we present two case studies discussing complete analysis applications, i.e. a tennis sports application (Chapter 4) and a surveillance application (Chapter 5). Both chapters employ 3-D modeling with one or two cameras. In the following, we briefly summarize the content of each chapter.

Chapter 1 introduces the background of human motion analysis and provides an overview of the related research. We introduce a generic behavior-analysis framework consisting of three processing layers: pixel, object and event-based analysis. Research requirements and contributions are presented. The chapter concludes with the thesis outline and scientific background of each chapter.

Chapter 2 first presents an overview of the state-of-the-art behavior-analysis techniques and analyze their merits and pitfalls. At the pixel-based processing level, background modeling techniques are presented. For object-based modeling, the existing approaches for human detection and tracking are discussed, such as background subtraction and temporal differencing, and mean-shift and model-based tracking. At the event level, typical techniques like Hidden Markov Models are discussed together with behavior description using temporal modeling. Finally, camera calibration techniques for analysis system are presented.

Chapter 3 presents the techniques for motion analysis of individual humans. Firstly, we propose a novel body representation to achieve posture

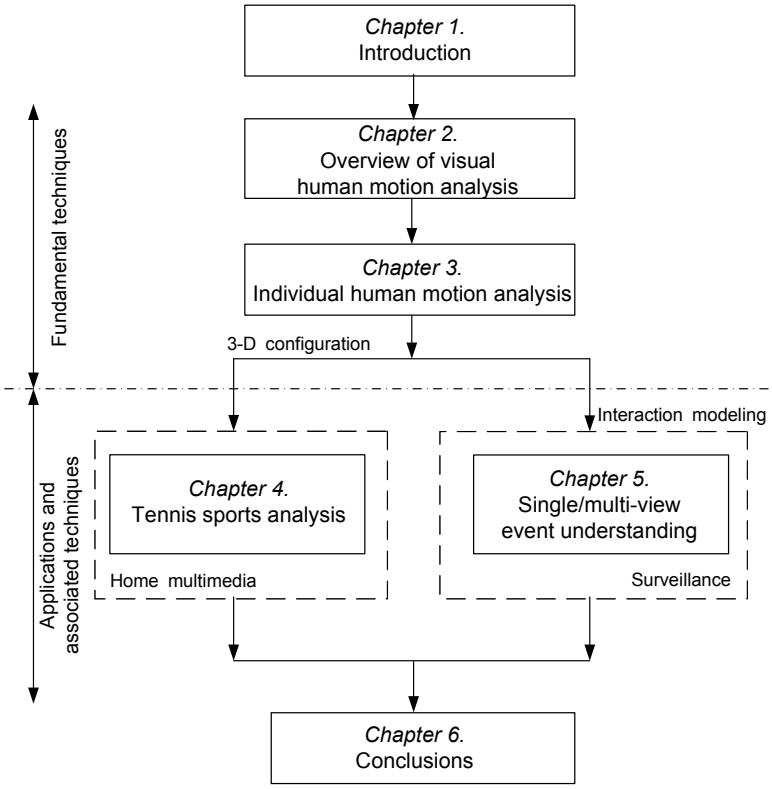

Figure 1.2: *Outline of the thesis structure.*

classification. Secondly, a novel body-part detection approach is presented. The color, body ratio are combined to provide clues for further reasoning, where the conventional assumption of up-right body posture is not required. The posture-classification results were first published in IEEE Proc. Int. Conf. Multimedia Modeling in 2007 [46] and later a full version was published in the SPIE Proc. Multimedia Content Access [47]. The body-part detection results were published in Springer Lecture Notes on Computer Science in 2008 [48].

Chapter 4 describes a complete case study that applies the behavior-analysis techniques in a sports video analysis application. First, the state-of-the-art work of sports video analysis is discussed. Afterwards, a video-based system for tennis sports analysis is presented, with details on tennis court calibration for 3-D camera calibration and background subtraction to find the players. Semantic inference is then discussed, involving the most relevant events for a tennis game. In the second half of the chapter, an alternative analysis system is presented based on audio clues and integration of game rules. Finally, the audio and visual clues are combined to facilitate tennis-

ball inference and tactical analysis. Our video-based analysis results were first published in the Proc. 26th Int. Symp. on Inform. and Comm. Theory in the Benelux [28] in 2005, the Proc. Int. Conf. Internet and Multimedia Systems and Applications (IMSA) [32] in 2006, and the IEEE Proc. Int. Conf. Consumer Electronics (ICCE) [29]. The complete system was published in the IEEE Trans. Consumer Electronics [30] in 2006. The audio-based analysis results were presented in Proc. IMSA [44] in 2006. The results combining video and audio clues were published in IEEE Proc. Benelux/DSP Valley Signal Proc. Symp. [45] in 2006.

Chapter 5 presents a surveillance application from monocular and multi-view video with our proposed behavior-analysis system using modified 3D modeling. This chapter firstly discusses the requirements for embedded surveillance applications. Then a new experimental *real-time* AV content-analysis system is introduced in detail. Afterwards, our behavior-analysis framework is proposed. Techniques for every step within the behavior-analysis framework are presented. We also extend the surveillance system to multiple-camera setting. Finally, the behavior-analysis framework is tested in several study cases in our experiments. Targeting at a bank-robbery detection, an initial system description was published in IEEE Proc. ICCE [50]. Later, the related results were published in SPIE Proc.VCIP in 2009 [51]. The behavior-analysis work addressed in this thesis was also embedded in a new experimental *real-time* AV content-analysis system. Its complete description was published in the IEEE Trans. Consumer Electronics [49] in 2009. The extension to multi-camera setup was published in the Int. Journal of Digital Multimedia Broadcasting [52] (special issue on video analysis, abstraction and retrieval: techniques and applications).

In *Chapter 6*, we conclude the thesis and indicate some future directions. The features and achievements involved in every chapter are summarized. In addition, we identify several interesting aspects that need further investigation.

Chapter 2

Overview of Visual Analysis of Human Behavior

2.1 Introduction

This chapter summarizes recent developments and existing techniques in the field of visual analysis of human behavior. The purpose of video understanding is that it should result in the recognition of events (either predefined by an end-user or learned by a system) in a given application domain (e.g., human activities). The involved processing starts with pixel analysis and ends with a symbolic description of what is happening in the scene. To reach this objective, several consecutive techniques have to be used, as the start of the processing is very different from creating symbolic descriptions at the end.

A typical behavior-analysis system is depicted in Figure 1.1 and consists of the following processing steps:

- *Pixel-based processing*, including background modeling, involving e.g. pixel labeling into histograms;
- *Object-based modeling*, including human detection and tracking;
- *Event-based analysis*, including event recognition and behavior understanding.

Each processing stage forms a sub-problem in general behavior analysis and facilitates a wide range of applications. The design of behavior-analysis system

has to satisfy system requirements, like operation speed, robustness, algorithm efficiency, classification performance, etc. We briefly discuss a few aspects here.

The robustness of a human-behavior analysis system can be expressed as the degree to which the system can maintain correct output decisions for large input variations. For example, the human detection should be able to correctly find humans when using unconstrained video inputs. The tracking part should follow the human correctly even for complex behavior patterns or fast movements. The final event recognition should give consistent results even when the persons shows a large variability in gender, age, clothing, illumination conditions, and so on.

The purpose of this chapter is to provide an overview of state-of-the-art techniques of each of the three levels of processing. The remainder of this chapter is organized as follows. First, the techniques for background modeling are presented in Section 2.2. Afterwards, Section 2.3 and Section 2.4 discuss different methods for human detection and human tracking, respectively. Tracking from multiple cameras is also described. Section 2.5 introduces the semantic analysis for behavior understanding and its description. Subsequently, we introduce the 3-D camera calibration in Section 2.6, which provides important information for the behavior-analysis system. Finally, Section 2.7 summarizes this chapter.

2.2 Background modeling

Updating of background models is an essential technique for human detection. The background models can be classified into 2-D models in the image plane and 3-D models in real-world coordinates. Generally, 2-D models have been explored for more applications due to their simplicity. If the camera locations are fixed, the key issue is to automatically recover and update background pixel sets from a moving video sequence. All elements in the captured scene background that have a time-varying nature complicate the robust performance of the background modeling. Examples of such elements are illumination variations, shadows, moving branches, etc., which pose many difficulties for the acquisition and updating of background pixels. Many algorithms have been published to handle such situations, like temporal averaging of an image sequence by Friedman [19], adaptive Gaussian estimation by Zivkovic *et al.* [110], and parameter estimation based on pixel processing, as proposed by Sun [95] and Grimson [23], etc. For example, Ridder *et al.* [23] model each pixel value with a Kalman Filter to compensate for illumination variations. Stauffer *et al.* [92] present a theoretic framework for recovering and updating background images, based on a process in which a mixed Gaussian model is used for each pixel value and online estimation is applied to update background images, in order to adapt to illumination fluctuations and disturbance in backgrounds.

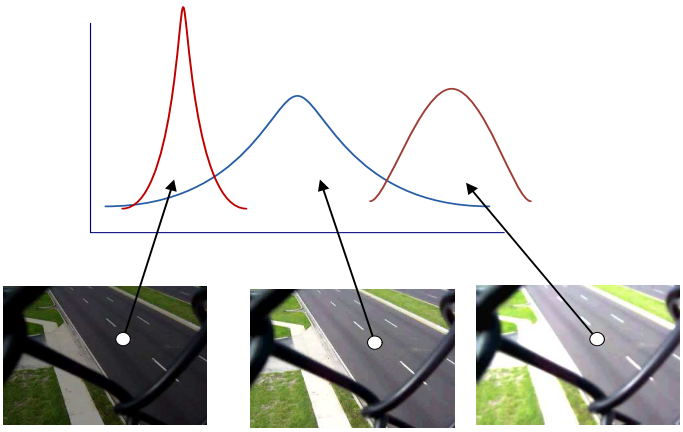

Figure 2.1: *An example of GMM-based background modeling. Each pixel is modeled with a set of Gaussian distributions.*

Toyama *et al.* [96] propose the Wallflower algorithm in which background maintenance and background subtraction are carried out at three levels: the pixel level, the region level and the frame level. Haritaoglu *et al.* [33] build a statistical model by representing each pixel with three values: its minimum and maximum intensity values, and the maximum intensity difference between consecutive frames observed during the training period. These three values are updated periodically. McKenna *et al.* [59] use an adaptive background model with color and gradient information to reduce the influences of shadows and unreliable color cues.

We have found that Gaussian Mixture Models (GMM) facilitate a fast operation while keeping satisfying accuracy. In addition, its memory requirements are feasible [92]. When using the GMM method, the following parameters of each Gaussian component need to be learned dynamically: the mean μ_k, variance σ_k^2, weight w_k of the k-th Gaussian, and the number of Gaussians K involved in the modeling. Then the Gaussian mixture distribution can be written as a linear superposition of Gaussians in the form

$$p(x) = \sum_{k=1}^{K} w_k N(x|\mu_k, \sigma_k^2). \tag{2.1}$$

An example of a GMM model ($K = 3$) is illustrated in Figure 2.1. Each pixel is modeled as a mixture of Gaussian distributions and any pixel intensity value that does not fit into one of the modeled Gaussian distributions is marked as a foreground pixel later. In this thesis, we adopt the GMM concept to model the background in a surveillance application (Chapter 5).

When a moving camera is used (e.g. sports video analysis in Chapter 4),

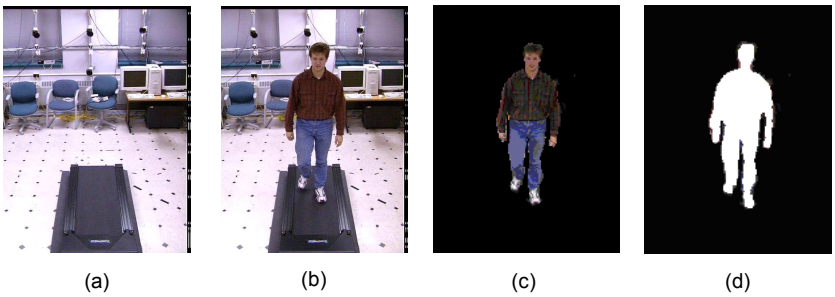

Figure 2.2: *Example of motion segmentation based on background subtraction: (a) background image; (b) current frame captured by the camera; (c) result of foreground object after performing per-pixel background subtraction; (d) silhouette image for the foreground object.*

the system has to align the input frame with the corresponding view of the background image prior to carrying out background subtraction (discussed in the next section). The motion parameters are computed directly relative to the background image, instead of first computing interframe motion. These motion parameters are subsequently used in the background subtraction algorithm to obtain motion-compensated camera views from the background image. Due to the uniform color for all tennis courts, a Gaussian model is sufficient to model the background within the court area. Because of its simplicity and suitability for real-time operation, this algorithm is adopted in our application of sports video analysis (Chapter 4).

2.3 Human detection

Human detection is aiming at segmenting the human body from images or video sequences. Human detection is an initial step for human tracking and behavior understanding. Human detection is generally composed of two steps: *motion segmentation* and *object classification*, which will be discussed below.

2.3.1 Motion segmentation

Motion segmentation in image sequences targets at detecting regions corresponding to moving objects such as vehicles and humans. Detecting moving regions facilitates later processing steps such as tracking and behavior analysis. At present, most segmentation methods use spatial, temporal information or combination of both types of information in the image sequence. Several conventional approaches for motion segmentation are outlined here.

- *Background subtraction*
  Background subtraction is a popular method for motion segmentation, especially in the case of a relatively static background. The fundamental idea is that moving regions are detected in an image by taking the pixel-by-pixel difference between the current frame $F_i(x, y)$ and the background image $B(x, y)$. The general processing steps of background subtraction are summarized as follows.

  1. Obtain the segmentation mask M from the pixels of frame F_i when the pixels satisfy

     $$|F_i(x, y) - B(x, y)| > T_h, \tag{2.2}$$

     where (x, y) denotes the actual pixel in the current frame and T_h is the threshold for differentiating the foreground and background pixels. Parameter T_h can be a fixed value or adaptive to a particular condition.
  2. Find the largest n blobs $B = (b_1, b_2, ..., b_n)$ in the segmentation mask M.
  3. Perform the dilate/erode operations to refine the n blobs by filling the holes for each blob $b_i, i = 1...n$.

  Figure 2.2 illustrates an example of motion segmentation based on background subtraction. Background subtraction is simple to implement, but sensitive to changes in dynamic scenes, e.g. light changes, moving leaves, etc. Therefore, the result highly depends on a good background model to reduce the influence of scene-background changes [60, 110].

- *Temporal differencing*
  Temporal differencing computes the pixel-wise differences between two or three consecutive frames in an image sequence to extract moving regions. Moving regions are detected in an image by taking the difference between the current frame $F_i(x, y)$ and the previous frame $F_{i-1}(x, y)$ in a pixel-by-pixel processing:

  $$|F_i(x, y) - F_{i-1}(x, y)| > T_h. \tag{2.3}$$

  Temporal differencing adapts to dynamic scenes, but it does not guarantee to extract all the relevant pixels, e.g., there may be holes left inside moving regions after performing temporal differencing. Therefore, a post-processing step is required to improve the performance. For example, moving targets are detected in video streams employing temporal differencing [56]. After the absolute difference between the current

and the previous frame is obtained, a threshold function is used to determine changes. By using a connected component analysis, the extracted moving sections are clustered into motion regions. This approach only works in particular conditions of the objects' speed and frame rate and is very sensitive to the threshold T_h.

- *Optical flow*
  Motion segmentation based on optical flow uses characteristics of flow vectors over time to detect moving regions in a video sequence. It tries to calculate the motion between two image frames which are taken at times t and $t + \delta t$ at every pixel position. Suppose a pixel at position (x, y) with intensity $I_t(x, y)$ moves by $\delta x, \delta y$ at time interval δt between two image frames, the following translation assumption is applied:

$$I_t(x, y) = I_{t+\delta t}(x + \delta x, y + \delta y). \tag{2.4}$$

  Optical-flow-based methods can be used to detect independently moving objects, even in the presence of camera motion. For instance, the displacement vector field is computed in [61] to initialize a contour-based tracking algorithm, called active rays, for the extraction of articulated objects. The results are used for gait analysis. However, most flow computation methods are very sensitive to noise. A more detailed discussion of optical flow can be found in Barron's work [2].

- *Hybrid methods*
  In addition to the basic methods described above, several other hybrid methods for motion segmentation have been reported. Using the extended Expectation Maximization (EM) algorithm, Friedman *et al.* [19] implement a mixed Gaussian classification model for each pixel. This model classifies the pixel values into three separate predetermined distributions corresponding to background, foreground and shadow. It also updates the mixed component automatically for each class according to the likelihood of membership. Hence, slowly moving objects are handled perfectly, while shadows are eliminated much more effectively. The authors of [11] have successfully developed a hybrid algorithm for motion segmentation by combining an adaptive background subtraction algorithm with a three-frame differencing technique.

Background subtraction is a straightforward approach to separate foreground objects from the scene. The obtained foreground objects maybe further analyze with respect to their features. For example, the silhouette feature is later used in this thesis to discriminate human behavior. The intrinsic simplicity of background subtraction is attractive for embedded system applications.

In our investigated cases, the obtained background model is generally based on the assumptions of limited illumination changes and stable scene background.

2.3.2 Object classification

Various moving regions in the video sequence correspond to different moving objects in actual scenes. For instance, the road-traffic video sequences include humans, vehicles and other moving objects, such as flying birds and moving clouds, etc. Prior to tracking of human objects and analysis of their behavior, it is essential to correctly classify moving objects. Object classification is a standard pattern-recognition task. Currently, there are three major types of existing approaches for object classification: *shape-based*, *motion-based* and *template-based* classification, which will be outlined below.

A. Shape-based classification
Different shape descriptions of motion regions such as contours, bounding boxes, silhouettes and blobs are utilized for classifying moving objects. For instance, Collins [11] utilizes various image-blob features, such as ranging from simple metrics like area, aspect ratio of the bounding box up to dispersedness in meaning. Moving-object blobs are classified into four classes (single human, vehicles, human groups and clutter) by using a viewpoint-specific three-layer neural network classifier. Singh *et al.* [90] derive directionality-based feature vectors from the silhouette contours and use the distinct data distribution of directional vectors for clustering and recognition. Temporal consistency constraints are considered in order to make classification results more precise. Lao *et al.* [51] use simple shape parameters of human silhouette patterns to distinguish humans from other moving objects.

B. Motion-based classification
The articulated motion from a non-rigid human body generally demonstrates one or more periodic patterns. This motion property is a strong cue for classifying moving objects. For example, the self-similarity measure described in [67] is periodic and a time-frequency analysis is applied to detect and characterize the periodic motion. Therefore, tracking and classification of moving objects are conducted using periodicity. In Lipton's work [56], residual flow is used to analyze rigidity and periodicity of moving objects. It is expected that rigid objects present little residual flow, whereas a non-rigid moving object such as a human being has a higher average residual flow and even shows a periodic component. Based on the above useful cues, human motion is distinguished from motion of other objects, such as vehicles.

C. Template-based classification
In template-matching techniques [46], the presence/location of humans is de-

termined, using predefined and manually encoded human templates representing specific human characteristics. Templates are usually parameterized by a specific function describing certain human features such as edges, contours or luminosity/color information. These templates are later compared with input images and regions of the image that are similar to the predefined template are classified as human. The main advantage of template-matching-based techniques is the fact that they are simple and easy to implement. However, most of the methods need to be initialized based on existing data in order to be effective. In addition, these methods are relatively restrictive and even deformable or flexible templates only allow for a small degree of variability.

In our investigated applications with human behavior analysis in this thesis, the shape information is important for further analysis (e.g. posture classification). Therefore, the motion-based method discussed in this section is not suitable due to its limitation of producing accurate human silhouettes. Also, it is complicated to enumerate templates for all possible poses and rotations. This constraint is similar to the one in knowledge-based face-recognition methods, where it is difficult to enumerate rules for all possible face poses and rotations. For these reasons, we concentrate on shape-based methods within this thesis and discuss their detailed implementation in Chapter 3.

2.4 Human tracking

Human tracking is the process of obtaining the trajectories of humans from a video sequence. Most existing human tracking techniques can be divided into the following approaches.

2.4.1 Region-based tracking

Region-based tracking is perhaps the most popular technique for performing the matching of objects over time. Most of these techniques rely on a Kalman filter (or equivalent) to perform the association between objects. In [13], the authors propose an efficient implementation of a multiple hypotheses tracker. They maintain all information about the associations between the currently tracked objects and the new observations until they are able to choose the correct associations (e.g. incoherent associations are removed). Each hypothesis (i.e. possible association) is processed with a Kalman filter. In [78], a Kalman filter is also used to effectively track interacting objects in a scene. However, this technique suffers from a problem of object initialization. In addition, based on the assumption of the Gaussian distribution, this technique is not conceived to handle multi-modal distributions of the state parameters. For instance, this technique is inadequate in dealing with the simultaneous presence

Table 2.1: *Key steps of the algorithm for mean-shift tracking.*

Given: the target model $\{\hat{q}_u\}_{u=1\ldots m}$ and its location $\hat{y}_0$ in the previous frame.

1. Initialize the location of the target in the current frame with $\hat{y}_0$, compute $\{\hat{p}_u(\hat{y}_0)\}_{u=1\ldots m}$, and evaluate the Bhattacharya coefficient

$$\rho[\hat{p}(\hat{y}_o), \hat{q}] = \sum_{n=1}^{m} \sqrt{\hat{p}(\hat{y}_o)\hat{q}_u}. \tag{2.5}$$

2. Derive the weights $\{w_i\}_{i=1\ldots n_h}$.
3. Find the updated location of the target candidate in the succeeding frame.
4. Compute $\hat{p}_u(\hat{y}_1)_{u=1\ldots m}$, and evaluate the Bhattacharya coefficient

$$\rho[\hat{p}(\hat{y}_1), \hat{q}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(\hat{y}_1)\hat{q}_u}. \tag{2.6}$$

5. While $\rho[\hat{p}(\hat{y}_1), \hat{q}] < \rho[\hat{p}(\hat{y}_0), \hat{q}]$
   Do $\hat{y}_1 \leftarrow \frac{1}{2}(\hat{y}_0 + \hat{y}_1)$
   Evaluate $\rho[\hat{p}(\hat{y}_1), \hat{q}]$.
6. If $\|\hat{y}_1 - \hat{y}_0\| < \epsilon$, then Stop iterating,
   Otherwise Set $\hat{y}_0 \leftarrow \hat{y}_1$ and go to Step 2.

of occlusions and a cluttered background resembling the tracked objects. Another solution consists of using a particle filter or more generally, probabilistic techniques [22]. For instance, Nummiaro *et al.*[69] use a color particle filter to track objects. The objects are modeled by a weighted histogram based on both the color and the shape of the objects. Then the particle filter compares the histograms of objects of frame $t-1$ and t at the sample positions in order to decide whether objects match or not. Summarizing, the region-based technique is robust to partial occlusions, is also rotation and scale invariant and computed efficiently. However, a limitation is that the tracker has difficulties to handle significant changes in object appearance.

Comaniciu *et al.* [12] use a weighted histogram computed from a circular region to represent the object. Instead of performing a full search for locating the object, they use the mean-shift algorithm. The mean-shift tracker maximizes the appearance similarity iteratively by comparing the histograms of the object, and the window around the hypothesized object location. Histogram similarity is defined in terms of the *Bhattacharya coefficient*, $\rho[\hat{p}(\hat{y}), \hat{q}]$. At each iteration, the mean-shift vector is computed such that the histogram similarity is increased. This process is repeated until convergence is achieved, which usually takes five to six iterations. For histogram generation, the authors use a weighting scheme defined by a spatial kernel, which assigns higher weights to the pixels closer to the object center. An outline of the mean-shift

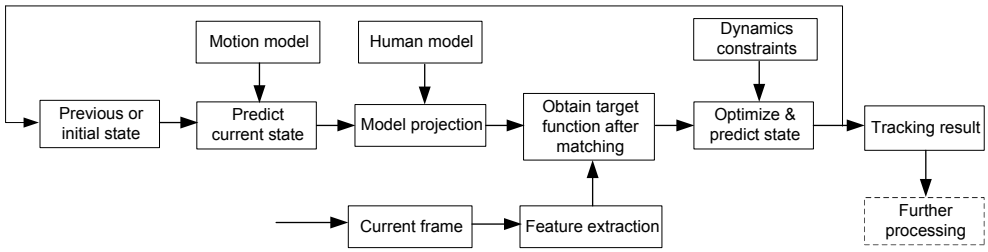

Figure 2.3: *Diagram for top-down model-based human tracking processing.*

tracking algorithm is presented in Table 2.1.

In our implementation, we intend to utilize the mean-shift tracking in our embedded application (e.g. surveillance application in Chapter 5) for two reasons:

- It has fast processing speed which facilitates real-time system operations;
- It can provide acceptable results (especially when the occlusion problem is not severe) with limited memory consumption, which satisfies the requirement of low-cost hardware design.

2.4.2 Model-based tracking

Compared to other methods, model-based tracking algorithms are often considered as sophisticated techniques which are difficult to implement. They also have some disadvantages, such as the necessity of constructing the models and a high computational cost. However, model-based tracking approaches usually perform better in complex situations (e.g. clutter, occlusions) [65, 75]. The general overview of involved steps in model-based human tracking is presented in Figure 2.3. During the tracking, the current state is predicted based on the motion model and the tracking result from the previous frame. The human model is projected into the image. Then the matching between features of the projected model and those of the current frame is performed to produce a matching function. Afterwards, the constraints for dynamic movements are modeled and integrated. The matching function is minimized by optimizing and/or correcting the predicted state. Finally, the state of human motion is obtained in the current frame.

For model-based approaches to capture human motion, the representation of the human body itself has steadily evolved from simple stick diagrams to 2-D contours and to 3-D volumes, as models become more complex. Some examples of different human-body models are shown in Figure 2.4. The models are briefly introduced in the following paragraphs.

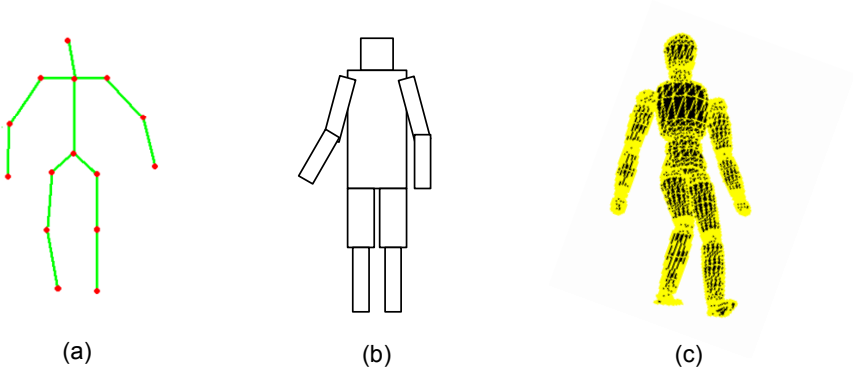

Figure 2.4: *Examples of human-body models: (a) Stick figure model; (b) 2-D contour model; (3) 3-D volumetric model.*

- *Stick figure models*
  The stick figure representation made of line segments is based on the observation that human motion is essentially the movement of the supporting bone structure. For example, Lee and Chen [53] use a model including 17 line segments and 14 joints to represent the features of the human head, torso, hip, arms and legs. It is assumed that the lengths of all rigid segments and the relative location of the feature points on the head are known in advance.

- *2-D contour models*
  The use of 2-D contours is directly associated with the projection of the human figure in images. For instance, Leung and Yang [54] apply a 2-D ribbon model to recognize poses of a human performing gymnastic movement. A moving edge detection technique is successfully used to generate a complete outline of the moving body. The technique significantly relies on image differencing and coincidence-edge accumulation.

- *3-D volumetric models*
  The above-mentioned 2-D models are only useful for specific camera viewing angles. Therefore, many researchers try to depict the geometric structure of the human body in more detail by exploring some 3-D volumetric models. In [105], the global motion of objects is tracked in 3-D using ellipsoid shape models. Robust results are reported on challenging sequences containing occlusions, shadows and reflections. However, the usage of 3-D models is restricted to impractical assumptions of simplicity regardless of the body kinematics constraints, and they have high computational complexity as well. In our investigated case of the surveillance application, volumetric models (highly realistic) are used to reconstruct the scene and provide the free-view representation of the humans in order

to enhance the event understanding.

2.4.3 Other tracking approaches

A. Feature-based tracking
The tracking of a set of features is another alternative solution for object tracking [21, 71, 62]. Typical features are centroids, perimeters, areas, color and appearance. For instance, Shi *et al.* [88] first compute the covariance matrix in a small image window. This window is shifted over the entire image in raster order. Afterwards, two eigenvectors corresponding to the two maximum eigenvalues of the covariance matrix are maintained. These eigenvectors define the directions of the maximum gradient and are thus used to detect corners (or T shapes) in the image. Compared to a region-based approach, feature-based tracking is robust to partial occlusions and can also be used to track crowds. However, its computational cost is the main disadvantage. In addition, the extraction of features is not robust, which often leads to tracking errors.

B. Active-contour based tracking
Active-contour based tracking algorithms track objects by representing their outlines as bounding contours and updating these contours dynamically in successive frames. These algorithms aim at directly extracting shapes of subjects and provide more effective descriptions of objects than region-based algorithms. For example, Peterfreund [74] explores an active-contour model, based on a Kalman filter for tracking non-rigid moving objects such as people in spatio-velocity space. In contrast with region-based tracking algorithms, active-contour based algorithms describe objects more effectively and reduce computational complexity. Even when disturbed by partial occlusions, these algorithms may track objects continuously. However, the tracking precision is limited at the contour level. Also, the recovery of the 3-D pose of an object from its contour is a demanding problem.

2.5 Event understanding and behavior description

For event understanding, it is required that we can distinguish the primary objects in the scene and classify what they are doing. The event understanding can be described with scenarios in which there is interaction between objects and the semantics are derived for individual behavior. The individual behavior semantics can be based on specific descriptors (e.g., position, speed, posture, trajectory and interaction). An event (e.g., abandoned bag, forbidden zone access, bank robbery) is either learned or predefined by an end-user. An event can be characterized by the involved objects, the individual behavior of objects, the interaction between objects, the initial time of the event recognition

and its duration. Event recognition is performed globally for the scene, captured by single or multiple cameras. Advanced event-based analysis should be able to not only handle the uncertainty associated to low-level features in order to maintain a high accuracy of event recognition, but also recognize complex events involving particular descriptors, such as the posture of an object.

2.5.1 Event understanding

Event-based analysis involves the detection and recognition of human behavior patterns, and the creation of semantic-level descriptions of human actions and interactions. In this thesis, understanding of human behavior is regarded as the classification of time-varying feature data. The process steps involved in the understanding are to learn the reference behavior from training samples, and then to devise both training and matching methods for coping effectively with small variations of the feature data within each class of behavior patterns. However, understanding human behavior is a complex and challenging task due to ambiguity caused by non-rigid body articulation, loose clothing and occlusion situations. The major existing methods for event understanding are now outlined.

A. Constraint-based approach
The constraint-based approaches recognize events in videos based on predefined event models in the form of templates, rules or constraints. For example, the use of templates has the advantage of conceptual simplicity and robust performance. They aim at matching an unknown test sequence with a group of labeled and pre-defined sequences representing typical human behavior. A constraint-based approach has been used recently in the matching of human movement patterns by Mori *et al.* [68]. In addition, Bobick *et al.* [5] use Dynamic Time Warping (DTW), a template-based dynamic programming matching technique, to recognize human gestures. Even if the time scale between a test sequence and a reference sequence is inconsistent, DTW can still successfully establish matching, as long as the time-ordering constraints are satisfied. In our recent work [52], we utilize the rule-based concept to detect a bank robbery. Both the individual person's spatial information and temporal interaction are considered in order to improve the detection accuracy of the system.

The constraint-based approach is relatively easy to implement with high efficiency. It is therefore generally applied in developing real-time systems [30]. Typically, a strong expert knowledge in a particular scenario is highly required for the generation of appropriate rules and the corresponding algorithm design. Therefore, a constraint-based approach cannot be tuned for a particular application in a straightforward way.

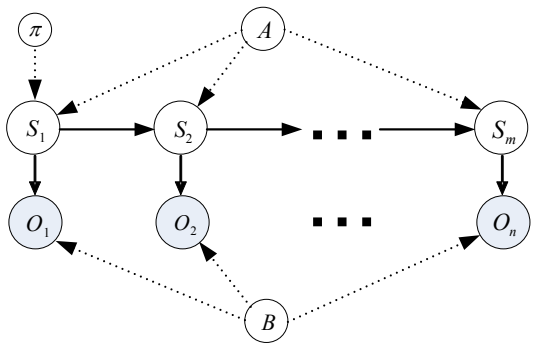

Figure 2.5: *An HMM model tied across time-slices. The parameters (nodes with outgoing dotted arcs) are explicitly represented.*

B. Learning-based approach

Learning-based approaches usually characterize the statistical description of human motion through a learning process. This approach is relatively robust to noise and invariant to changes in time intervals from input image sequences. Statistical techniques are applied to automatically learn event models based on training data [109]. The Dynamic Bayesian Network (DBN) is generally adopted to represent a statistical model. Then both certain events and behaviors are distinguished by analyzing time sequences and their statistical models. For example, Remagnino *et al.* [84] describe interactions between objects using a two-layer Bayesian network. The method relies on pixel-level recognition supported by motion analysis, and involves high-level concepts for events and scenarios. Finally, the relationships between these concepts are modeled and concepts are associated with each other.

Hidden Markov Models (HMMs) are widely used as an important learning-based approach [70, 7]. An example of an HMM model is visualized in Figure 2.5. Suppose an HMM has m states $S = \{s_1, s_2, ..., s_m\}$ and n observation symbols $O = \{o_1, o_2, ..., o_n\}$. The HMM is fully specified by the triplet $\lambda = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$. Let the state at time step t be S_t, then the $m \times m$-state transition matrix $\mathbf{A}$ can be defined by

$$\mathbf{A} = \{a_{ij} | a_{ij} = P(S_{t+1} = s_j | S_t = s_i)\}, \quad 1 \leq i, j \leq m. \tag{2.7}$$

The $m \times n$-state output probability matrix $\mathbf{B}$ is defined as

$$\mathbf{B} = \{b_j(k) | b_j(k) = P(o_k | S_t = s_j)\}, \quad 1 \leq j \leq m, 1 \leq k \leq n. \tag{2.8}$$

The initial state distribution vector $\boldsymbol{\pi}$ is specified as

$$\boldsymbol{\pi} = \{\pi_i | \pi_i = P(S_1 = s_i)\}, \quad 1 \leq i \leq m. \tag{2.9}$$

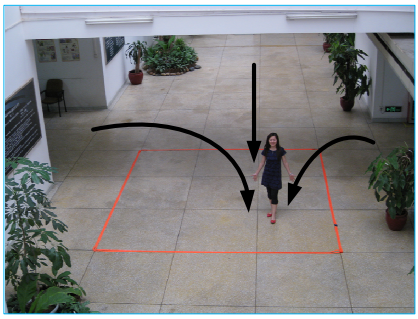

Figure 2.6: *Example of using an HMM model to recognize the human behavior: three classes are defined to determine the person's trajectory.*

For event classification, an HMM model is assigned to each of the predefined classes for the observation. Each HMM is trained based on the Baum-Welch algorithm [81]. The learning process can calculate all parameters λ of the model using the training data. Given an observation sequence $Obv = \{Obv_1, Obv_2, ..., Obv_q\}$, we can calculate $P(Obv|\lambda_p)$, which is the probability of the observation sequence Obv given model p with λ_p. After calculating the output probability of each model, the model with maximum probability is chosen as the recognition result. We can therefore recognize the class C_{max} as being the one that is represented by the maximum probable model among K types:

$$C_{max} = \arg\max_{p} P(Obv|\lambda_p), \ 1 \leq p \leq K. \tag{2.10}$$

Let us now provide an example to explain how to use HMM models to classify different events. Three individual HMM models are trained for recognizing three different types of human behavior, e.g. from which door the person enters the monitoring scene (see Figure 2.6). The observation feature vector is formed by $[x_1, y_1, v_1]^T$, where x_1, y_1, v_1 are the person's x-coordinate, y-coordinate and movement speed v_1 in the image, respectively. Then the training process can calculate all parameters of the model using the training trajectory data. When applying the model to perform behavior analysis, we can calculate the likelihood of the given trajectory that is generated from the HMM model. In this example, the corresponding HMM parameters are: m=8, n=3, and K=3. Finally, the trajectory direction is classified into one of the following types: *from left, from right, from above.*

Summarizing, HMMs allow a more sophisticated data analysis with spatio-temporal variability. During the training stage, the number of states of an HMM is specified, and the corresponding state transition and output probabilities are optimized. Therefore, the generated symbols can correspond to the

observed image features of the examples within a specific movement class. At the matching stage, the probability with which a particular HMM generates the test symbol sequence, corresponding to the observed image features, is computed. In our implementation, an extended model of HMM, called Continuous Hidden Markov Model (CHMM) with left-right topology is adopted as an effective posture classifier (see Chapter 3).

As an alternative learning-based approach, neural network-based systems are commonly applied to behavior understanding for unconstrained object motions. Johnson *et al.* [40] describe the movement of an object in terms of a sequence of flow vectors, each of which consists of individual components representing the positions and velocities of the object in the image plane. Afterwards, a statistical model of object trajectories is formed. Sumpter *et al.* [94] introduce a new neural network structure which has a smaller scale and faster learning speed, and thereby results in a more effective prediction of object behavior.

The syntactic-based (grammatical) approach has also been used for visual behavior recognition. Brand [6] uses a simple non-probabilistic grammar to recognize sequences of discrete behavior. Ivanov *et al.* [37] describe a probabilistic syntactic approach to the detection and recognition of temporally extended behavior and interactions between different agents.

The above learning-based methods either model single-person events, or require *a-priori* knowledge about the number of people involved in the events. Variation in data may require complete re-training, so as to modify the model structure and parameters to accommodate those variations. Furthermore, there is no straight-forward method of expanding the domain to other events, once training has been completed. If more events are added to the current domain, or if we want to model events in a new domain, the existing models have to be re-trained, using the new data and the model structure has to be re-defined for the new events.

C. Clustering-based approach

Clustering-based approaches do not explicitly model events but utilize clustering techniques for event detection. Therefore, a large amount of available training data is important for robust performance. However, this condition sometimes restricts its applicability to embedded systems if only limited training data is available. The clustering-based methods of event detection include spatio-temporal derivatives [104] and co-embedding prototypes [107]. Both methods find event segments by partitioning a spectral graph of the weight matrix. The weight matrix is estimated by calculating a heuristic measure of similarity between video segments. These methods assume maximum length of an event and are restricted to a single person or a single threaded event detection. Rao *et al.* [82] propose human action recognition using spatio-temporal curvatures of 2-D trajectories. Their method initiates without any

Table 2.2: *Human action description and corresponding vocabulary sets.*

Set notation for human action:
The universe set of human action: U
U={action|action=<agent-motion-target>}
agent set: S
S={s_i|s_i=various body parts as agent term}
={head, torso, arm, leg}
motion set: V
V={v_j|v_j=movement of the body part}
={stay, move left, move right, raise, lower, stretch, withdraw}
target set: O
O={o_k|o_k=the other person's body parts}
={head, torso, left arm, left leg, right arm, right leg}

event model and forms clusters of similar events based on their spatio-temporal curvatures. Their method is also restricted to single person event detection, but it makes no assumptions about the length of an event, and the event representation which is based on spatio-temporal curvature is also view-invariant.

In our application studied in this thesis, domain knowledge is highly required for the design of real-time embedded system implementation of those applications. For example, the tennis-game rules provide strong constraints in tennis sports analysis. They can be effectively modeled and the training step is not required at all. Therefore, a constraint-based approach is an effective solution at the event-based level for this application. However, in the object-level modeling, a learning-based approach is generally adopted as it can produce a robust classification result. For example, the HMM model proves to be a useful tool to classify posture types, which will be presented in detail in Chapter 3.

2.5.2 Description of behavior

It is important to describe human behavior in a natural language, which is suitable for many applications, e.g. video surveillance. For example, Ryoo *et al.* [85] develop a novel representation scheme, employing context-free grammar, which uses natural language to describe visual scenes.

Statistical models are commonly used for behavior description and they interpret certain events and behavior by analysis of time sequences and statistical modeling. For example, Remagnino *et al.* [84] describe interactions between objects using a two-layer agent-based Bayesian network. These meth-

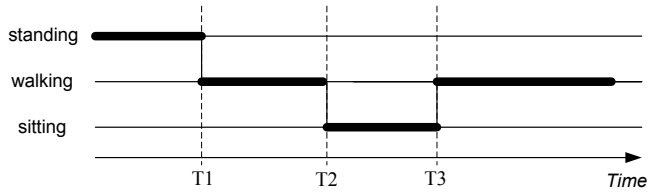

Figure 2.7: *Example of a temporal description for human behavior.*

ods utilize lower-level recognition based on motion concepts, and do not yet involve high-level concepts, such as events and scenarios, and the relationships between these concepts. Such concepts need high-level reasoning based on a large amount of *a-priori* knowledge. Recently, Kojima *et al.* [43] propose a new method for generating natural language descriptions of human behavior appearing in real image sequences. First, a person's head region is extracted from each frame, and the 3-D pose and position of the head are estimated using a model-based approach. Next, the head motion trajectory is divided into the segments of monotonous movement. The conceptual features for each segment, such as degrees of pose changes, position and the relative distances from other objects in the surroundings, are evaluated. Meanwhile, the most suitable verbs and other syntactic elements are selected. Finally, the natural language text for interpreting human behavior is generated by machine-translation technology.

The human actions can be defined as <agent-motion-target> according to the linguistic theory of 'verb argument structure' [1]. The argument structure of a verb allows us to predict the relationship between the syntactic arguments of a verb and their role in the underlying semantics. Table 2.2 shows an example of human action description and its corresponding vocabulary sets. However, the above description cannot represent the temporal information about individual behavior or interactive behavior between different people. Therefore, it is necessary to find a suitable tool to model the temporal relation. A simple model representing the order of various motion types (standing, walking, sitting) is visualized in Figure 2.7. The time instant for the start and end of each person's motion type is indicated.

Although there is some progress in the description of behavior, several key issues remain unsolved. For example, it is difficult to properly represent semantic concepts, to map motion characteristics to semantic concepts, and to choose efficient representations to interpret the meanings of a video scene. This leaves important ground for significant research contributions in this area.

Even when the correct interpretation of the scene has been performed, the scene-understanding system needs to enable its understanding to different users (experts of the application domain, end-users like operators, police, etc.). Furthermore, the *a-priori* knowledge of the application domain requires for-

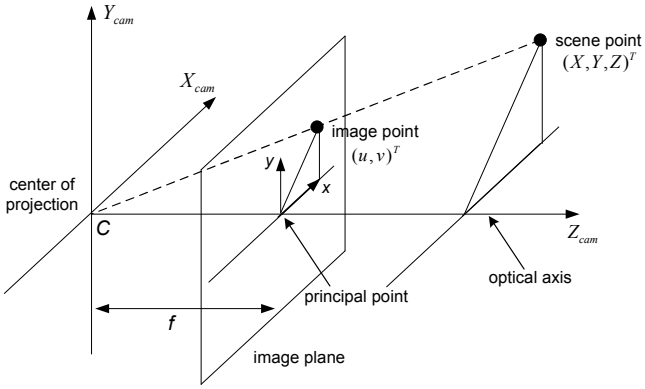

Figure 2.8: *The ideal pinhole camera model presents the relationship between a 3-D point $(X, Y, Z)^T$ and its corresponding 2-D projection point $(u, v)^T$ onto the image plane.*

malization to enable the domain experts to describe their knowledge and their methodology for analyzing the scene. In scene understanding, there are four main types of knowledge to represent: (1) the empty scene of the surrounding (e.g. its geometry), (2) the sensors (e.g. calibrated and synchronized cameras), the communication network and the processing units, (3) the physical objects expected in the scene (e.g. 3-D model of a human) and their dynamics (location and speed), and (4) the events of interest for end-users.

In our description of a surveillance application, XML files are generated and further adapted to the ViPER file format (see Appendix A.2). Then, the multiple-person behavior and the associated semantic events are described, according to spatial and temporal relationships defined by domain knowledge. Furthermore, the timing configuration is included in this description. The above method can effectively provide an integrated representation of the event. Also, the annotations can be transmitted over the network in order to exchange information between the processing units of a large and distributed surveillance system. Chapter 5 presents such a networked event-recognition system.

2.6 Camera calibration

Camera calibration can be an essential component of a human-behavior analysis system, particularly in those cases where the analysis system is used to compute actual behavior in terms of trajectory distance, speed, etc. Its role is to estimate the metric information of the camera. In other words, camera calibration attempts to establish the relationship between the camera's internal coordinate system and the coordinate system of the real world. It is therefore the first step for a calibrated motion capture system.

Let us now explain the underlying principles for camera calibration. First, we introduce the projection of points from 3-D space onto a 2-D image plane. This projection operation is based on the idealized model of a pinhole camera, which is a close approximation to most real cameras. We derive the projection equations and formulate them, using the the homogeneous coordinates framework (see Appendix A.1).

The *pinhole camera model* is widely used in computer vision. The model projects the 3-D real world to the 2-D space. The setup for the perspective camera model is illustrated in Figure 2.8. A point of the 3-D scene object is projected along the ray from the camera center to the object point. The intersection of the ray with the image plane defines the position in the image. In a real camera, the image plane is actually behind the camera center, and the image is projected onto it up-side down, but for simplicity, we assume that the image plane is in front of the camera. This is equivalent, but it relieves us from considering many minus signs in the formal statement of the problem.

Let us assume a camera with the optical axis being collinear to the Z_{cam}-axis and the optical center being located at the origin of a 3-D coordinate system (see Figure 2.8). The other two axes (X_{cam}- and Y_{cam}-axis) are in a plane perpendicular to the Z_{cam}-axis. The two axes of the coordinate system of the captured images are often assumed parallel to the x and y axes of the camera frame if optical distortion is ignored. Let a 3-D point in the camera frame be $(X, Y, Z)^T$ and its correspondent image point be $(u, v)^T$. The relationship between these two points is written as:

$$u = \frac{Xf}{Z} \quad \text{and} \quad v = \frac{Yf}{Z}, \tag{2.11}$$

where f is the focal length of the camera lens. To avoid a non-linear division operation, the previous relation can be reformulated using the projective geometry framework, as

$$(\lambda u, \lambda v, \lambda)^T = (Xf, Yf, Z)^T. \tag{2.12}$$

This relation can be expressed in matrix notation by

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{2.13}$$

where $\lambda = Z$ is the homogeneous scaling factor.

In the sequel, we provide an example of using camera calibration techniques to implement 2D-3D mapping.

The geometric transformation is required to map points in the image to real-world coordinates on the ground. Since both the ground and the displayed image are planar, it is a plane-to-plane transformation. In addition,

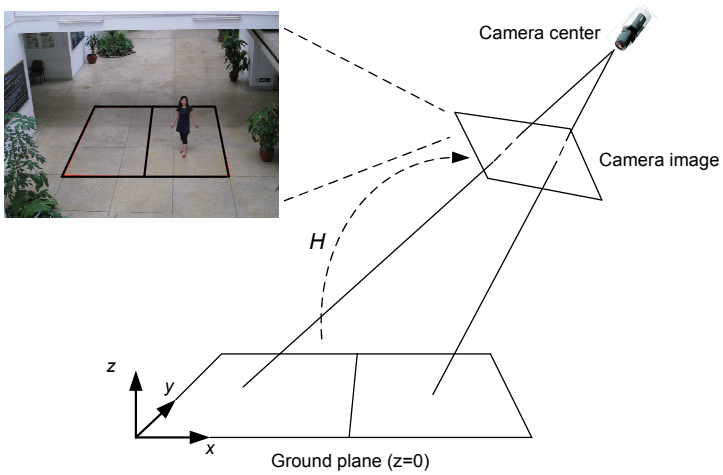

Figure 2.9: *Example of the corresponding homography based on camera calibration.*

we analyze the human behavior based on the person trajectory and/or speed on the ground, so that the height information of the human is not required. Without loss of generality, we can place the ground plane at $z = 0$ and obtain the 2D-3D mapping by a geometric transformation, specified by

$$p' = Hp = \underbrace{\begin{pmatrix} f & 0 & o_x \\ 0 & f & o_y \\ 0 & 0 & 1 \end{pmatrix}}_{\substack{\textit{internal camera} \\ \textit{parameters}}} \underbrace{\begin{pmatrix} r_{00} & h_{01} & r_{02} & t_x \\ r_{10} & h_{11} & r_{12} & t_y \\ r_{20} & r_{21} & r_{22} & t_z \end{pmatrix}}_{\substack{\textit{camera rotation,} \\ \textit{translation}}} \begin{pmatrix} x \\ y \\ z = 0 \\ 1 \end{pmatrix}. \quad (2.14)$$

This is a homography, represented by the 3×3 transformation matrix H (see Figure 2.9). The principal point $(o_x, o_y)^T$ is located at the center of the image (Figure 2.9) and t_x, t_y, t_z denote the camera translation along the X-, Y-, Z-axis, respectively. The matrix H transforms a point $p = (x, y, w)^T$ in real-world coordinates to image coordinates $p' = (x', y', w')^T$. Because matrix H is scaling invariant, eight free parameters have to be determined. They can be calculated from four reference points whose positions are both known in the ground model and in the image.

In a similar way, it is possible to transform the image data into a 3-D model space. This 3-D scene reconstruction is a useful tool in semantic-event analysis, which may be utilized in multimedia applications [26]. The accurate and realistic reconstruction in a virtual space can significantly contribute to the scene understanding, like crime-evidence collection and tactical analysis.

This approach may be of interest for advanced surveillance applications, such as home-care monitoring and robbery-detection surveillance.

In this section, we have shown that a mapping of the 3-D data into the 2-D image data can be realized with a projective transform between actual scene and captured images, based on an ideal pin-hole camera model. In Chapter 4, we utilize a standard tennis-court model to obtain the correspondence between the real-world domain and image domain. Similarly in Chapter 5, a pattern grid with four white lines is used to map the correspondence between the reference points detected in the image and the points in the actual scene. Afterwards, the exact displacements of a person can be obtained. For example, the location and speed of humans in the scene are calculated prior to the semantic analysis. In addition, 3-D reconstruction is performed to enhance the scene visualization.

2.7 Summary and conclusions

This chapter has presented an overview of techniques for visual analysis of human behavior. Three levels of techniques have been discussed within a general processing framework: pixel-level processing, object-level modeling and event-level analysis.

- In the pixel-level processing, background modeling with pixel labeling is briefly introduced, thereby distinguishing foreground objects from the background of the scene. We have found that Gaussian Mixture Models are attractive for fast algorithm deployment, while keeping satisfying accuracy.

- In the object-level modeling, various techniques of human detection and tracking are discussed. Human detection is composed of two steps: motion segmentation and object classification. Four types of human tracking methods are discussed: region-based, feature-based, active-contour-based and model-based tracking. From these types, region-based processing is adopted for further experiments in the upcoming chapters. Region-based tracking is feasible when occlusions are not severe and it facilitates a real-time system implementation.

- In the event-level analysis, the current event-understanding approaches can be classified into three types: constraint-based, learning-based and clustering-based methods. In our applications, the domain knowledge plays an important role in the analysis so that the constraint-based approach seems most beneficial for system implementation. For sub-event analysis as in object-level modeling, a learning-based approach is attractive because it gives a robust classification result.

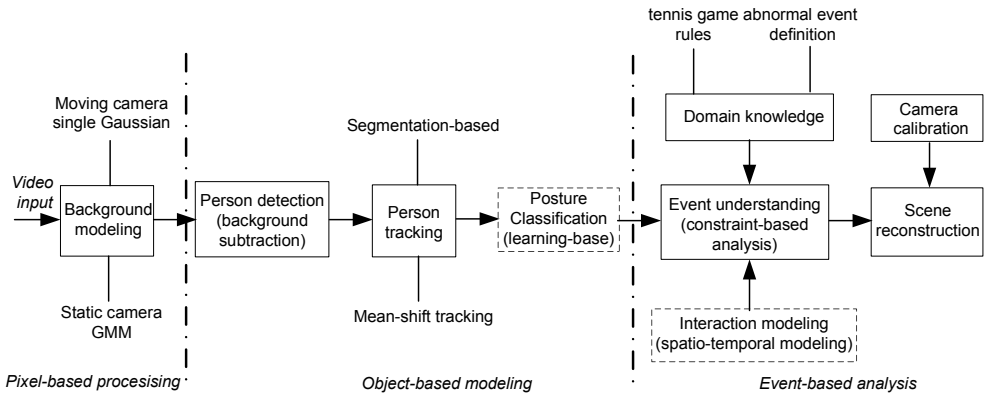

Figure 2.10: *Block diagram of processing steps involved in Chapter 4 and Chapter 5.*

- Finally, camera-calibration techniques are presented. Camera calibration enables the 2D-3D mapping and provides a platform for normalized motion configuration (i.e. location and speed) and scene visualization.

In the literature, current approaches have been designed under different assumptions and for different purposes. Each technique alone is not sufficient to address the variety of all possible situations and to be selected as an acceptable solution regarding the complexity of the video-understanding problem. Therefore, delicate attention is payed for choosing suitable algorithm in order to achieve fast operation.

This chapter has presented an overview of existing techniques involved in different steps of human behavior analysis. In this thesis, we focus on developing a fast and robust embedded system. We have also discussed the algorithm selection at each processing step in our implementation of motion analysis. The techniques discussed in this chapter are connected with each other as depicted in the generalized diagram of Figure 2.10. This diagram has been elaborated and worked out for two application studies, which are presented in Chapter 4 and Chapter 5. The diagram indicates which parts involve pixel-, object-, and event-level processing steps. This diagram applies to both applications, but for some functions, there are differences which are specific for the individual application. The background modeling is different for indoor and outdoor scenes. Also, advanced event analysis of human behavior requires that specific functions are added to the person tracking to facilitate a better behavior analysis. Examples of such specific functions are human skeleton modeling and posture classification, which will be discussed in Chapter 3.

Chapter 3

Motion Analysis of the Human Action

3.1 Introduction

The previous chapter has presented the state-of-the-art work within the human motion analysis. The motion analysis is constrained to the behavior of a single person. However, finding the moving areas in the video is not sufficient for interpreting the human behavior. Research increasingly touches upon semantic analysis through the object-based processing of the individual human actor. For this purpose, this chapter adds the classification of a human posture, so that the orientation of the human body and its limbs can be used for interpretation. This addition forms a good basis for performing interaction modeling, activity recognition in more detail. The combination of both steps is essential for object/scene analysis and behavior modeling of deformable objects. This chapter discusses these techniques in more detail.

The main issue in human motion analysis is the lack of a sufficiently accurate measurement of e.g. the contour of the body and position of individual body parts. The body-part detection and classification of human posture would play an important role in performing the activity recognition and interaction modeling, leading to object/scene analysis and behavior modeling of deformable objects. Some previous work used multiple cameras [66, 65] to obtain more reliable observations. However, most of the computer-vision-based approaches require more than one camera and highly accurate camera-calibration parameters as well. Unfortunately, these approaches are not always

feasible for surveillance and e.g. low-cost consumer multimedia applications. Therefore, we concentrate on using a single camera only, but instead add a posture classification of human body that will allow us to distinguish the human behavior.

This chapter is divided as follows. We first present an approach for automatic posture classification based on a novel silhouette descriptor in Section 3.2. Then a novel scheme for body-part detection and 2-D skeleton construction is introduced in Section 3.3. Finally, Section 3.4 draws conclusions and summarizes the chapter.

3.2 Individual posture classification

Accurate detection and efficient recognition of various human postures are very useful for understanding the scene. Existing posture-representation techniques can be generally classified into two categories: local feature-based and global feature-based methods.

Local feature-based approaches use local silhouette [87] or spatio-temporal features such as Scale-Invariant Feature Transform (SIFT) [57] and 3-D Hessians [24] to represent human activity in a video. In the silhouette-based methods, human activity can be regarded as a temporal process in which human silhouettes continuously change over time. If the extracted feature in each frame characterizes the human silhouette, temporal variations of these features will implicitly characterize motion kinematics. However, an important limitation of the aforementioned approaches is that they do not incorporate global characteristics of the activity and the performance relies on the foreground segmentation, especially when the background is complex and changing.

Global feature-based approaches use global features such as optical flow to represent the state of motion in the whole frame at a time instant. With static background, one can represent the type of motion of the foreground object by computing features from the optical flow. To avoid articulated tracking or segmentation, recent work has shifted towards the combination of Histogram of Oriented Gradients (HOG) and optical flow [10]. The advantage of these techniques is their relatively simple implementation. However, they are significantly affected by the large variance of body postures and clothing.

In silhouette-based human action recognition, dimensionality reduction techniques are adopted to project the original data covering a space with a high number of dimensions into data having a much lower number of dimensions. The most popular reduction methods are Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) [16] and Locality Preserving Projections (LPP) [35]. In the recent sparse learning methods, Shao et al. [86] propose the Spectral Regression Discriminant Analysis (SRDA) for dimensionality reduction.

In some investigated cases (e.g. surveillance with luminance pictures), only silhouette configuration is available to analyze human behavior. Therefore, we have to rely on a silhouette-based approach. In this section, we discuss a silhouette-based approach, based on a novel body representation, to facilitate the posture classification in an efficient way.

3.2.1 Posture representation

Prior to conducting the temporal modeling scheme of a Continuous Hidden Markov Model (CHMM) to recognize the posture type, we propose a new, yet simple and effective shape descriptor, called HV-PCA, to represent the silhouette in each frame. Let us now explain this new descriptor.

Firstly, every detected person silhouette is scaled to an $M \times N$ sub-image for normalization, where we choose $M = 180$ and $N = 80$. Then within the template, we apply the horizontal and vertical projections which will be explained in the sequel. Because each individual projection (horizontal or vertical) is non-orthogonal, the projections along the vertical and horizontal axis in the pixel domain with 180 and 80 dimensions are redundant. Therefore, Principal Component Analysis (PCA) is used to obtain a more compact and still accurate representation in each frame. In the vertical projection, the 180-D shape vector is divided into three parts and thus a feature matrix of 60×3 dimensions is obtained for each frame. Then the size of each feature matrix is reduced to 2×3 after performing PCA. Afterwards, the previous reduced matrix is rewritten into a 6×1 vector. Although there are different options for the matrix form, we adopt the division scheme of 60×3 to achieve the balance between computation cost and recognition rate. Similarly, the horizontal matrix of 20×4 is reduced to a vector of 8×1 for the horizontal projection, and divided into four parts. Thus, both dimensions lead to a 14-D vector to represent the human silhouette. In summary, the principal formal algorithm steps of HV-PCA are defined as follows.

Suppose (x, y) represents every pixel that belongs to a silhouette within a shape template S, its value Sil is represented as

$$Sil(x, y) = \{ \begin{array}{ll} 1, & \text{if } (x, y) \text{ belongs to foreground,} \\ 0, & \text{otherwise.} \end{array} \tag{3.1}$$

Then we can calculate the horizontal projection $H(m)$ in the m-th column and vertical projection $V(n)$ in the n-th row in the frame I by

$$H^I(m) = \sum_{j=0}^{M-1} Sil(m, j), \quad 0 \leq m \leq N - 1, \tag{3.2}$$

and

$$V^I(n) = \sum_{i=0}^{N-1} Sil(i,n), \quad 0 \leq n \leq M-1. \tag{3.3}$$

Finally, we can obtain a 14-D feature vector of the silhouette $\mathbf{Obv}^I$ in frame I by using PCA, hence

$$\mathbf{Obv}^I = (P_M(H^I(.)), P_N(V^I(.)))^T, \tag{3.4}$$

where $P(.)$ indicates our part-based PCA implementation (4-part horizontal projection and 3-part vertical projection). Then every $\mathbf{Obv}^I$ is set as observation input to the Continuous Hidden Markov Model (CHMM) classifier for parameters learning and testing.

3.2.2 Temporal modeling with CHMM

Due to noise from segmentation errors, a single-frame recognition is not sufficiently accurate when we require general motion classification. For a good posture recognition, temporal consistency is required.

Hidden Markov Models (HMMs) are a popular probabilistic framework for modeling processes that have structure in time. They have clear Bayesian semantics, efficient algorithms for state and parameter estimation, and they automatically perform dynamic time warping. An HMM is essentially a quantization of a system's configuration space into a small number of discrete states, together with probabilities for transitions between those states. A single finite discrete variable indicates the current state of the system with the index number. Any information about the history of the process needed for future inferences must be reflected in the current value of this state variable.

We adopt the HMM as our posture classifier, since it has proven to be an effective tool for sequential data processing. We use the Continuous Hidden Markov Model (CHMM) with left-right topology [81]. Suppose a CHMM has E states specified by the set $Q = \{q_1, q_2, ..., q_E\}$ and F output symbols in set $V = \{v_1, v_2, ..., v_F\}$. The model is fully specified by the triplet $\lambda = \{A, B, \pi\}$. Let the state at time step t be s_t, then the $E \times E$-state transition matrix $\mathbf{A}$ can be defined by

$$\mathbf{A} = \{a_{ij} | a_{ij} = P(s_{t+1} = q_j | s_t = q_i)\}, \quad 1 \leq i, j \leq E. \tag{3.5}$$

The $E \times F$-state output probability matrix $\mathbf{B}$ is defined as

$$\mathbf{B} = \{b_j(k) | b_j(k) = P(v_k | s_t = q_j)\}, \quad 1 \leq j \leq E, \ 1 \leq k \leq F. \tag{3.6}$$

The initial state distribution vector π is specified as

$$\pi = \{\pi_i | \pi_i = P(s_1 = q_i)\}, \quad 1 \leq i \leq E. \tag{3.7}$$

We assign a CHMM model to each of the predefined posture types for the observed human body. Each CHMM is trained based on the Baum-Welch algorithm [81]. The learning process can calculate all parameters of the model using the training data. In other words, the triplet λ is obtained for each model. After having the models for each posture, we can proceed to implement the online testing. Given an observation sequence $Obv = \{Obv_1, Obv_2, ..., Obv_T\}$, we can calculate $P(Obv|\lambda_i)$, which is the probability of the observation sequence Obv given model i with λ_i. The probability $P(Obv|\lambda_i)$ can be obtained by using the forward algorithm [81]. After computing the probability of each model output, the model with maximum probability is chosen as the recognition result. We can therefore recognize the posture class C_T as being the one that is represented by the maximum probable model among K types:

$$C_T = \arg\max_i P(Obv|\lambda_i),\ 1 \leq i \leq K. \tag{3.8}$$

In our investigated case (T=30, K=5), every given posture is finally classified into one of the following types: *left-pointing, right-pointing, squatting, raising hands overhead* and *lying*. The background of this choice of postures refers to the case study that will be discussed in Chapter 5. In that case, we use a limited set of posture classifications to interpret the posture when the persons are standing. In this section, we are simply interested in the detection performance of the posture-classification algorithm.

Table 3.1: *The comparison of different shape-based features in posture classification.*

Posture type	Skeleton [20]	A-Skeleton [76]	4-Hu [55]	HV-PCA
Left pointing	82%	88%	84%	92%
Right pointing	78%	84%	88%	90%
Squatting	56%	60%	78%	84%
Lying	56%	64%	80%	76%
Raising hands overhead	66%	72%	84%	86%

3.2.3 Experimental results

We have performed our posture classification using various single-person action (left pointing, right pointing, squatting, lying and raising hands) at video-capturing rate 15 frames/s. We have used 200 video sequences (40 for each posture type) for training and 500 sequences (100 for each posture type) for

testing. The posture-classification results using several existing shape-based features are summarized in Table 3.1. The ground truth data is obtained manually. We have calculated the classification accuracy to measure the performance of different methods. The accuracy is obtained by calculating the ratio of correct posture-type sequences detected ζ_l and the number of testing sequences ν_l for each posture type l, which is denoted as

$$Accuracy = \frac{\zeta_l}{\nu_l} \times 100\%. \tag{3.9}$$

From Table 3.1, it is noted that our proposed HV-PCA feature achieves an average accuracy rate of about 86% and outperforms other proposals.

We have also compared our result to state-of-the-art methods in Table 3.2. These methods are all tested on the same public dataset Weizmaan [4]. Although our result is not the best compared with other methods, our proposed algorithm enables real-time operation while others do not claim their efficiency in fast implementation.

Table 3.2: *The comparison of the state-of-the-art features tested in Weizmaan dataset.*

Jia et al. [39]	Zheng et al. [106]	Shao et al. [86]	HV-PCA
90.9%	98.8%	100%	93.2%

3.3 Body-part detection

Successful pose estimation and human-body modeling facilitate the semantic analysis of human activities in video sequences [64, 47]. If the body parts (face and hands) cannot be accurately detected when self-occlusion problem occurs or no color information is available, the approach of using skin-color model [47] is not an ideal solution. We need to look for other approaches for effective body-part detection.

Accurate detection and efficient tracking of various body parts are ongoing research topics. However, the computation complexity needs significant reduction to obtain a real-time performance, especially for surveillance applications. Existing fast techniques can be classified into two categories: appearance-based and silhouette-based methods.

- *Appearance-based* approaches [97, 73] utilize the intensity or color configuration within the whole body to infer specific body parts. They can simplify the estimation and collection of training data. However, they are significantly affected by the variances of body postures and clothing.

- For the *silhouette-based* approach [20, 33, 102, 76], different body parts are located employing the external points detected along the contour, or internal points estimated from the shape analysis. The geometric configuration of each body part is modeled prior to performing the pose estimation of the whole human body. However, the highly accurate detection of body parts remains a difficult problem, due to the effectiveness of segmentation. Human limbs are often inaccurately detected because of the self-occlusion or occlusion by other objects/persons.

Summarizing, both silhouette and appearance-based techniques do not offer a sufficiently high overall accuracy of body-part detection. Also, the assumption of upright posture is generally required.

To address the challenging problem of accurately detecting and modeling human-body parts in a fast way, we contribute in two aspects. First, various differentiating body features (e.g. body ratio, shape, color) are integrated into one framework to detect different body parts without the assumption of the human's upright posture. Second, we have proposed a novel scheme for capturing human motion, that combines the trajectory-based estimation and body-based modeling. This is effective to improve the detection accuracy. As our system is efficient and achieves nearly real-time performance (around 10 frames/second), we facilitate its application in a surveillance case study.

The structure of this section is as follows. Section 3.3.1 briefly presents the scheme. Section 3.3.2 introduces every detection component involved. The body-part detection that is based on seamless integration of different observation clues, is explained in detail. Promising experimental results and analysis are presented in Section 3.3.4.

3.3.1 System architecture of body-part detection

When combining the trajectory-based estimation and body-based detection, we intend to capture the human motion and locate the body parts using a skeleton model. The block diagram of our proposed scheme is shown in Figure 3.1. First, each image covering an individual body is segmented to extract the human silhouette after shadow removal. Second, both the trajectory-based and body-based modules are co-operating based on a particular sequence of internal functions. The position of the moving object in every frame is extracted. Occurring situations (behaviors) can be validated along the estimated trajectory for every individual person. Based on the trajectory-based estimation, the system initializes the local body-part detection. In this body-modeling module, various features are applied, such as appearance, body ratio and posture direction. Furthermore, the center point of the whole body is extracted. After different body parts are detected, the human geometry is modeled. Finally, the skeleton model of every person is constructed.

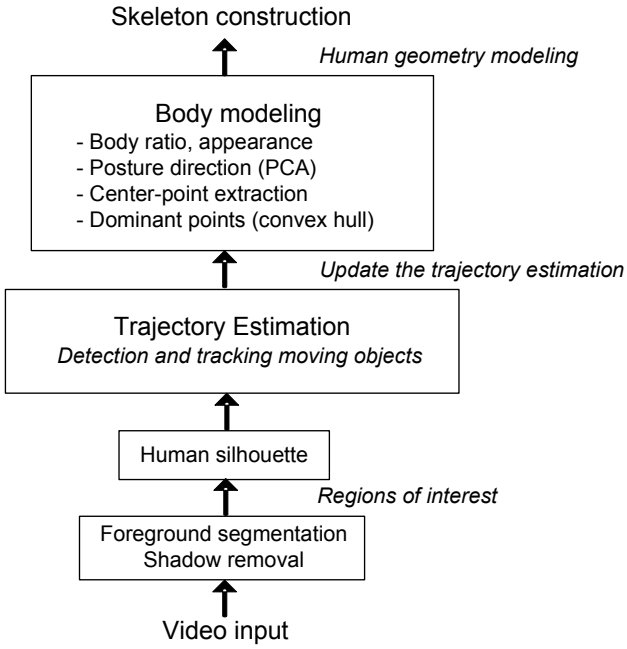

Figure 3.1: *Block diagram of our body-part modeling system.*

3.3.2 Component algorithms of body-part detection

A. Background substraction

Background modeling is generally the first step of detection and/or analysis of moving objects in a video sequence. We perform an adaptive background subtraction to support person-behavior analysis. The intention is to maintain a statistical background model at every pixel.

In the case of common pixel-level background subtraction, the scene model has a probability density function for each pixel separately. A pixel from a new image is considered to be a background pixel if its new value is well described by its density function. For a static scene, the simplest model could be just an image of the scene without the intruding objects. After the background modeling, the next step would be to e.g. estimate appropriate values for the variances of the pixel intensity levels from the image, since the variances can vary from pixel to pixel. Pixel values often have complex distributions and more elaborate models are needed. The Gaussian Mixture Model (GMM) is generally employed for the background subtraction. We apply the algorithm from [110] to produce the foreground objects using a Gaussian-mixture probability density. The parameters for each Gaussian distribution are updated in a recursive way. Furthermore, the method can efficiently select the appropriate number of Gaussian distributions during pixel processing, in order to fully

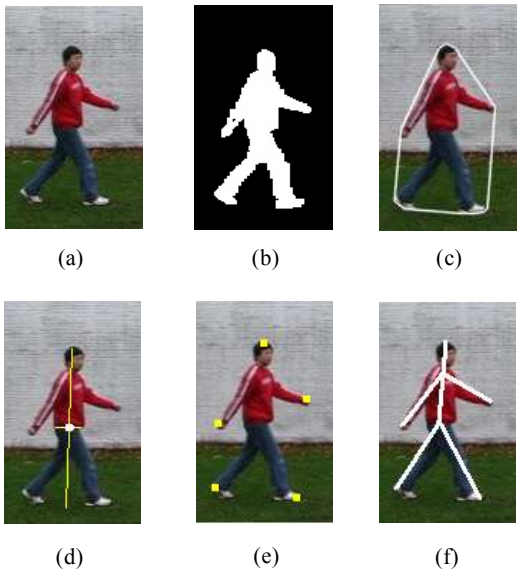

Figure 3.2: *Procedure of body-based processing: (a) original frame, (b) foreground segmentation (after shadow removal), (c) body modeling based on convex hull, (d) center-point estimation, (e) body-part location and (f) skeleton construction in single-person motion.*

adapt to the observed scene.

In the actual segmentation of foreground and background, shadow removal is another important issue. Based on the assumption that shadows decrease the brightness of pixels but do not affect their color, shadows are detected and removed [110]. To consider changes in illumination during the process of video acquisition, the pixels labeled as background are used to update in a recursive manner. Finally, the labeled foreground pixels are grouped together to represent potentially moving objects.

B. Trajectory estimation

The trajectory-based module estimates the human position over time, i.e. the movement, which is regarded as a fundamental function of surveillance systems. In our trajectory-based module, we apply blob tracking in two approaches. In a simple setting (e.g. static background, no occlusion), the first approach is based on an object's segmented binary mask. In the second approach, we employ the broadly accepted mean-shift algorithm for tracking persons, based on their individual appearance model, which is represented as a color histogram. When the mean-shift tracker is applied, we detect every new person entering the scene and calculate the corresponding histogram model in the image domain. In subsequent frames for tracking that person, we shift the

person object to the location whose histogram is the closest to the previous frame. After the trajectory is obtained, we can conduct the body-based analysis at the location of the person in every frame.

C. Body-based modeling

The body-based processing block models the human motion by a skeleton model. The detailed procedure is illustrated in Figure 3.2. In the example of single-person motion, the input frame (Figure 3.2(a)) is segmented to produce a foreground blob, after shadow removal is applied(Figure 3.2(b)). Then the convex hull is implemented for the whole blob (Figure 3.2(c)). The dominant points along the convex hull are strong clues, in the case of single-person body-part detection. They infer the possible locations of body parts, like head, hands and feet. Here we employ a *content-aware* scheme (Section 3.3.3) to estimate the center point (Figure 3.2(d)), which is fundamentally used to position the human skeleton model. Meanwhile, dominant points along the convex hull are selected and refined (Section 3.3.3) to locate the the head, hands and feet (Figure 3.2(e)). Finally, different body parts are connected to a predefined skeleton model involving a center point, where the skeleton is adapted to the actual situation of the person in the scene (Figure 3.2(f)).

3.3.3 Construction of 2-D skeleton model

We represent the body by using a skeleton model, which is used to infer the relative orientation of body parts and body posture. The center point is first estimated from the silhouette. Afterwards, it is connected to different body parts to construct the skeleton model.

A. Center-point extraction

The center point plays an important role in the skeleton model as a reference point. Its estimation accuracy significantly affects the detection of body parts. Here we apply a *content-aware* scheme to detect the center point c_i at the frame with index i. Contents of posture direction, human-body ratio and appearance are taken into account. The key processing steps are illustrated in Figure 3.3.

The human posture's direction can be estimated by the major axis m_i of the body's foreground region at the frame i. The major axis is determined by applying *Principal Component Analysis* (PCA) to the foreground pixels. Its direction is given by an Eigenvector v associated with the largest Eigenvalue of its covariance matrix. Along the above direction and based on the somatological knowledge, we initially classify the whole body into three segments: head, upper body (including torso and hands) and lower body (two legs). Also, an initial body boundary b_i, dividing upper body and lower body, is produced. Next, within a neighboring area A from body boundary b_i, we perform the

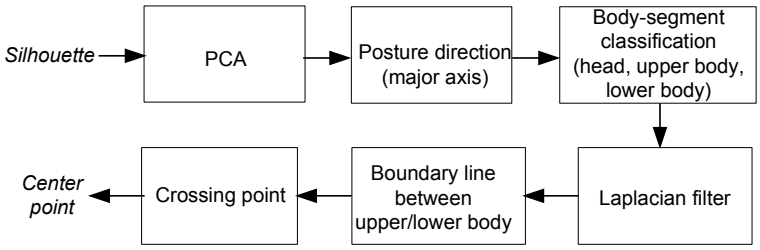

Figure 3.3: *Block diagram of center-point extraction.*

Laplacian filter $L_i(x, y)$ to each pixel (x, y) prior to a thresholding function $f(a, \delta)$ using threshold δ. If $L_i(x, y) > \delta$, then the thresholding $f(a, \delta) = 1$. Otherwise, $f(a, \delta) = 0$. Then we search the optimal boundary line b'_i between the upper body and lower body in Equation (3.10) by

$$b'_i = \arg\max_{b_i} \sum_{(x,y)\epsilon b_i} f(L_i(x, y), \delta), \tag{3.10}$$

where $L_i(x, y)$ indicates the Laplace operation with the 3×3 kernel at point (x, y). Finally, the center point C_i is located by the crossing point of the major axis m_i and the boundary line b'_i in Equation (3.11), hence

$$C_i = m_i \odot b'_i, \tag{3.11}$$

where "$\odot$" denotes returning the intersection position between two lines. During our experiments, we have found that this center-point extraction is effective and accurate, and it is superior to the Centroid-of-Gravity (CoG) approach of the whole blob, as used in [20]. An example is visualized in Figure 3.4. Our proposed scheme is simple but effective, even when disturbed by residual noise after shadow removal. If the clothes between the upper body and the lower body are similar in the appearance, only the silhouette feature is employed. The center point is estimated based on the domain knowledge of the human-body ratio.

B. Skeleton-model extraction

After the center point is obtained, a skeleton model of the human body is extracted. The key processing steps are illustrated in Figure 3.5. Different body parts are connected to the center point according to a predefined human geometry model, which is similar to the one reported in [76]. Every individual part is estimated according to the Euclidean distance between the center point C_i and every dominant point along the convex hull at the frame i. Based on the body-ratio knowledge, we initially select a set of dominant points P_i with the maximum distance in the three body segments, i.e. head, upper body and

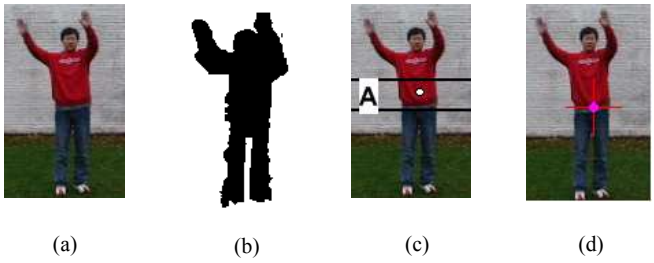

Figure 3.4: *Estimation of center point: (a) original frame, (b) silhouette after foreground segmentation, (c) result of CoG approach, (d) result of* content-aware *center point.*

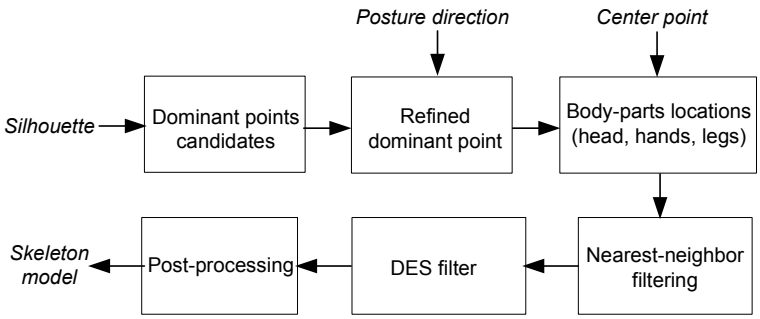

Figure 3.5: *Block diagram of skeleton-model extraction.*

lower body. These dominant points are used to infer the locations of potential body parts. As we obtain the body segments (head, upper body, lower body) along the posture direction, we can refine the points P_i in each individual segment to locate the body parts. Then we use a simple nearest-neighbor filtering scheme to correlate different body parts over time. Afterwards, a Double Exponential Smoothing (DES) filter is added to refine the results. This filter provides a good performance for moving object tracking [29].

The DES smoothing operator is defined by

$$\begin{cases} s_i = \alpha \cdot o_i \ + \ (1-\alpha) \cdot (s_{i-1} + d_{i-1}) \ , \\ d_i = \gamma \cdot (s_i - s_{i-1}) \ + \ (1-\gamma) \cdot d_{i-1} \ , \end{cases} \tag{3.12}$$

where o_i is the observed body-part position value at the frame i. The parameter s_i refers to the position after smoothing the observed position, d_i represents the trend of the change of body-part position, and α and γ are two weighting parameters controlling motion smoothness. Equation (3.12) applies to every detected body-part position for the individual person. The first smoothing equation adjusts s_i directly for the trend of the previous period with d_{i-1}, by adding it to the last smoothed value s_{i-1}. This helps to eliminate possible

position discontinuities. The second smoothing equation updates the trend, which is expressed as the weighted difference between the last two position values.

After the smoothing filter is performed on the observed body parts, another post-processing step is implemented to improve the detection accuracy. If the distance between the detected hands and the center point is below a predefined threshold, we set the location of the hands as a default value, i.e. the position of center point. This additional processing can remove some inaccurate observations and improve the accuracy, especially in the self-occlusion case.

Table 3.3: *Comparison of the detection accuracy of three methods for five human actions.*

		Method of [20]	Method of [76]	Our method	Frames
Walking	Feet	78%	85%	90%	800
	Hands	70%	78%	84%	
	Head	93%	96%	100%	
Leaping	Feet	90%	90%	94%	400
	Hands	71%	76%	83%	
	Head	95%	96%	100%	
Pointing	Feet	93%	95%	99%	400
	Hands	90%	94%	96%	
	Head	97%	100%	100%	
Kicking	Feet	90%	93%	98%	400
	Hands	88%	92%	95%	
	Head	98%	99%	100%	
Falling	Feet	73%	75%	82%	600
	Hands	58%	65%	77%	
	Head	88%	90%	98%	

3.3.4 Experimental results and discussions

In our experiments, we have tested the algorithm for different monocular video sequences covering 2,600 frames. The video sequences are recorded at 15-Hz frame rate with a resolution of 320 × 240 samples (QVGA). The sequences cover different persons, background, clothes and behaviors in both indoor and outdoor situations.

We have evaluated our scheme with different activities such as walking, pointing, kicking, leaping and falling. We have implemented two well-known contour-based methods [20, 76] for performance comparison. Table 3.3 presents a confusion matrix showing the accuracy comparison using different methods,

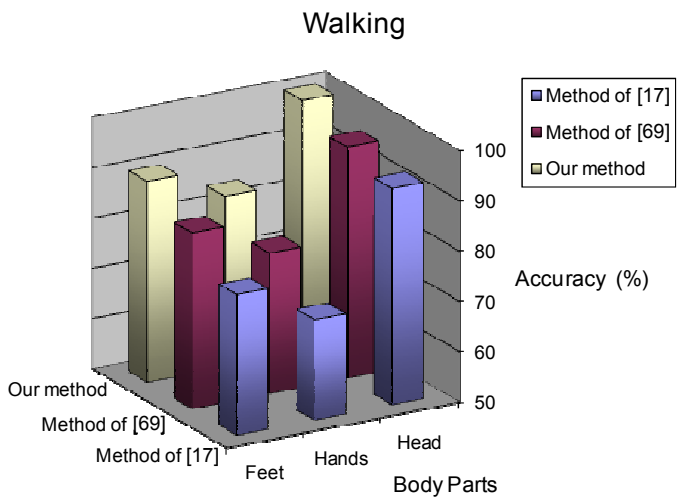

Figure 3.6: *Comparison of the detection accuracy of three different methods for walking.*

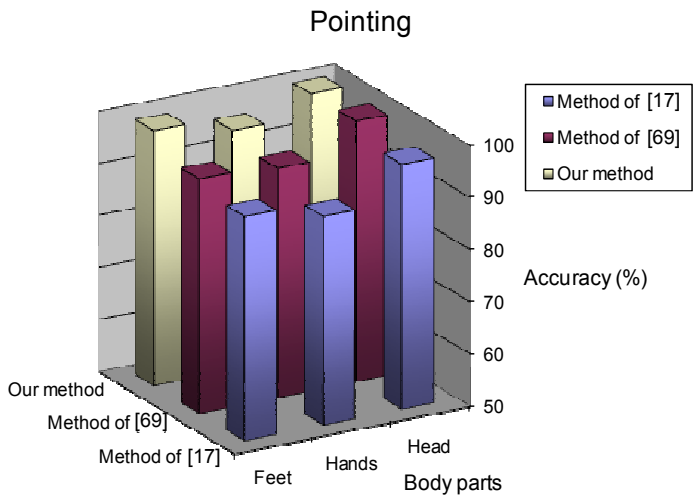

Figure 3.7: *Comparison of the detection accuracy of three different methods for pointing.*

and testing frames for each action. The accuracy is obtained by calculating the ratio of frames with correctly detected human body and the number of testing frames. Figure 3.6-Figure 3.10 present the accuracy comparison when using different methods. In our experiments, the ground truth of body-part locations are manually obtained. The maximum tolerable errors in the evaluation is set to 15 pixels. Our system is implemented in C++ on a 3.0-

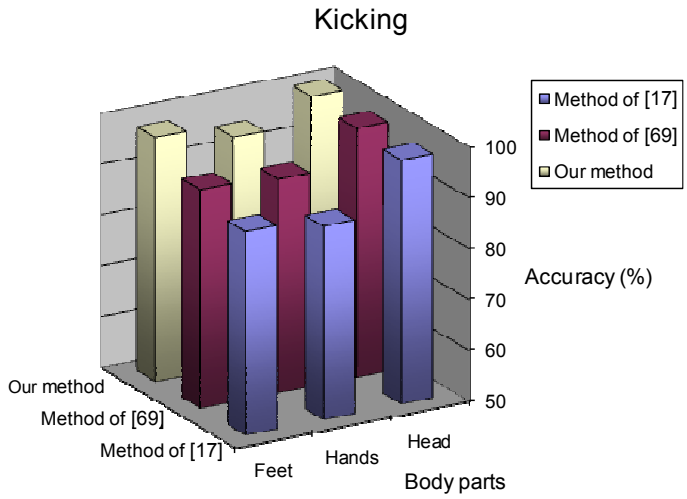

Figure 3.8: *Comparison of the detection accuracy of three different methods for kicking.*

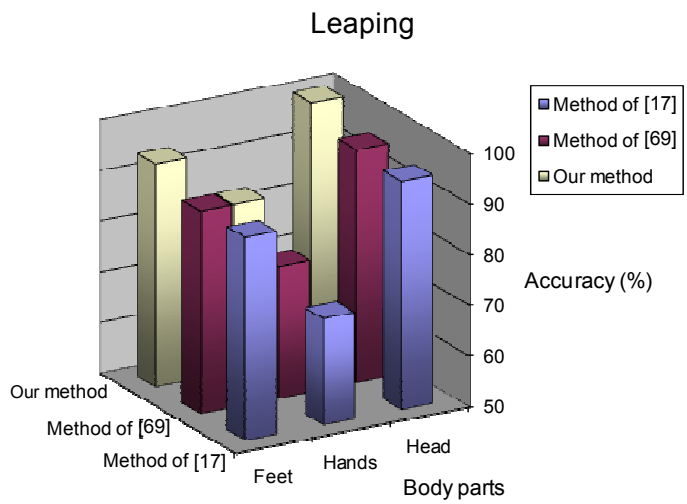

Figure 3.9: *Comparison of the detection accuracy of three different methods for leaping.*

GHz PC. The detection system operates nearly at real-time speed (around 10 frames/second).

From our experiments, we have found that the dominant points (with high curvature) along the contour play an important role in the three presented contour-based methods. If the dominant points are well visible, e.g. in the postures of pointing and kicking, all three methods yield similar performance.

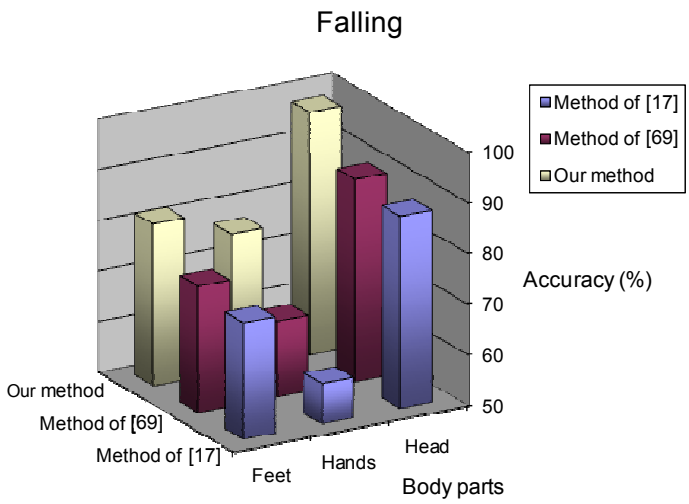

Figure 3.10: *Comparison of the detection accuracy of three different methods for falling.*

However, as we integrate the temporal constraints by employing the DES filter, our detection accuracy is higher by around 5%, especially in the case of the self-occlusion when the hands/legs appear within the silhouette. Another interesting point is that our method does not assume that the human posture is upright. Moreover, the posture direction can be estimated in our algorithm. In the case of falling, our method clearly outperforms the other two [20, 76] by approximately 20% in the detection of hands.

3.4 Summary and conclusions

This chapter has introduced motion analysis for individual human actions. The current efforts focus on generating a general framework while discussing the associated object-based techniques. We aim at keeping the balance between accuracy and effectiveness.

Firstly, we have presented a fast posture-classification scheme. Our proposed HV-PCA descriptor with temporal modeling achieves an accuracy rate of about 86% for posture recognition and outperforms other existing proposals. As our human motion scheme is efficient and achieves a fast performance (around 10 frames/second), it enables a surveillance system or further analysis of human behavior. The occlusion problem is not yet thoroughly tackled at the current stage. To further combat the problem of occlusion, multiple cameras will be employed for capturing the same scene from different angles, and it is necessary to integrate an effective occlusion-handling module, which was reported in [31] to improve the motion-analysis robustness.

Secondly, we have proposed a novel approach for body-part detection, that combines trajectory-based estimation and body-based analysis in a co-operating way, to capture the human motion and locate different body parts. The trajectory-based module provides a platform for performing body-based analysis. The body-based module updates the tracking process, and describes the body geometry efficiently by a 2-D skeleton model. We have presented a new algorithm for accurately locating the body center point, using the body silhouette and an upper/lower-body separation line. This algorithm outperforms the conventional center-of-gravity approach from existing literature, addressing the same center-point usage. Body-part detection was performed after estimation of the center point, analysis of body ratio, silhouette and appearance. An advantage is that the conventional assumption of upright body posture is not required. The above scheme has proven to be a fast (nearly real-time speed at 10-Hz frame rate) and effective technique for the automatic detection of different body parts within monocular video sequences in indoor/outdoor areas.

However, the current system has a few limitations. The self-occlusion problem is not completely solved, requiring additional exploration, as the dominant points along the convex hull fail to differentiate and locate the underlying body parts within the silhouette. We have found that the color appearance of the person is important in the case of self-occlusion. The region-based nature of color can be utilized to improve the body-part segmentation. Also, it is useful to capture motion sequences from different viewpoints and train the optimal parameters for various activities, thereby aiming at becoming more view-independent in performance. Most of the above limitations are addressed in the case study in Chapter 5 to come to an improved performance.

Up to this point, we have discussed the object-based processing (e.g. body-part detection, posture classification) for intermediate-level analysis of human motion. To obtain semantic meaning in specific scenarios, for example sports video analysis, both the tracking and body-based analysis require specific domain knowledge, such as the game rules. Furthermore, the individual motion analysis can upgrade to conduct interaction modeling, when more than one person is involved in the scene. In the next chapters, the motion analysis is further investigated to obtain semantic analysis in different applications. In Chapter 4, the study case of tennis sports is elaborated by employing the techniques of individual motion analysis. In Chapter 5, a study case of visual surveillance is investigated with interaction modeling. In both chapters, a 3-D scene reconstruction is performed to achieve semantic-level analysis. Specific domain knowledge (e.g. the tennis game rules) is integrated in the system design and algorithm selection.

Chapter 4

Tennis Sports Analysis Application

4.1 Introduction

4.1.1 Motivation

The previous chapters have discussed several fundamental techniques for human motion analysis. We are now going to apply a selection of those techniques in different applications in this and the subsequent chapter. This chapter concentrates on a consumer application, i.e. tennis sports analysis, where we further add other aspects, such as domain knowledge and 3-D camera calibration to obtain real-time semantic analysis.

The relevance of a consumer case involving video analysis, is motivated by the continuous growth in media storage capacity and the handling of large databases in multimedia systems of audio/video (AV) files for movies and music. For these databases, various applications such as sports video (tactics) analysis and object-based manipulation are potentially attractive. These applications will facilitate practical user requirements involving effective database management, indexing, and quick searching and retrieving of specific contents. With the advent with increasing of hard-disk capacity running to 1 TBytes, large video databases for consumer applications are gradually developing. The large storage capacity of compressed video on disks increases the need for fast storage and retrieval functions, realizing quick user-friendly searching for and access to specific parts of the video data. The identification

of those parts in the video can be improved or enhanced by metadata, describing the specific properties of key objects in the video scene. Such metadata should then be generated by a tool, which is active at recording time. This chapter explores such an analysis tool for finding rich content and presents an experimental application for home use. In consumer television, sports videos constitute a major percentage of the total video content, provided by public and commercial television channels. Because of the growing offer of TV channels providing full coverage of large sports events, we have focused on sports video analysis and finding meaningful parameters and event data.

4.1.2 State-of-the-art of sports video analysis

Content understanding of sports video is an active research topic, in which the past research can be roughly divided into four stages. Earlier publications [42, 72] have only focused on pixel and/or object-level analysis, which segment court lines and/or track the moving players and the ball. Evidently, such systems may not provide the semantic interpretation of a sports game. The second generation of sports video analysis exploits the general features to extract highlights from the sports video. Replayed slow-motion [75], density of scene cuts and sound energy [98] are common input features used by such systems. Although this highlight-based analysis is more general than earlier proposals, many systems still lack sufficient understanding of a sports game, as a viewer cannot deduce the whole story from looking to a special event only. The third stage of sports analysis is an event-based system [42, 72, 98], aiming at extracting predefined events in a specific sports genre. Visual features in the image domain, such as object color, texture and position, are useful clues broadly adopted by these systems. Despite these approaches yield acceptable results within the targeted domain, it is hard to extend the approach of one sports type to another, this is because the condition of video-capturing and scene organization are highly dependent on the sports type. In the fourth stage of sports analysis, research shows an increasing interest for constructing a generic framework for sports video analysis. Opposite to the highlight-based system [41], the previous systems [17, 93] try to recognize more events with rich content by modeling the structure of sports games. A predefined event can be identified based on the model generated during a training phase, which studies the interleaving relations of different dominant scene classes. This type of approaches has been proven to be applicable to multiple sports genres. Unfortunately, the primary disadvantage is that it does not take the behavior of key objects into account, thereby failing to provide sufficient events or tactical meaning.

Among the above sports analysis systems, there are several algorithms that are highly related to our work. Zhou *et al.* [108] propose a tennis video analysis system approaching a video retrieval application. It detects the court

lines and tracks the moving players, then extracts the events, such as base-line rally, based on the relative position between the player and the court lines. Improved work is presented in [55], where the authors further upgrade the court detection and player tracking algorithms. Park [73] first defines four types of camera views in tennis video, involving global, medium, close-up, and audience shots, and then detects events like first-service failure in terms of the interleaving relations of these four views. In the system described in [41], sports video is characterized by its predictable temporal syntax, recurrent events with consistent features, and a fixed number of views. A combination of domain knowledge and supervised machine learning techniques is employed to detect the different event boundaries. Mikic [66] also employs shot-based modeling, but creates a concept of mid-level representation to bridge the gap between low-level features and the semantic shot class.

In this chapter, the semantic analysis of sports is achieved by integrating domain knowledge of games. More spefically, we aim at developing a real-time sports-analysis system by designing fast algorithms at each processing level.

4.1.3 Requirements for home-use sports analysis systems

Although several sports video analysis algorithms aiming at different types of sports have been proposed, sports analysis for home use is not easily achieved, because the system should be robust for different users. The users may apply these algorithms in various ways. We require that the home-use application involves a system that provides a broad range of analysis at different semantic levels. Besides the robustness, for a multitude of users, the platform for executing such an analysis system should be efficient and not too expensive.

In order to fulfill the above requirements, we present a fully automatic and real-time system, which grows in the direction of multi-level analysis of tennis video sequences. The main contributions of our research are in three aspects. First, we employ a 3-D camera model to bridge the pixel-level, object-level and scene-level of the tennis sports analysis, which enables to deliver various semantic results for different users. Second, we employ the combination of visual cues in the real-world domain to classify events. Third, audio signals are utilized to perform racket-hit detection and increase robustness and more detail in event understanding. Afterwards, automatic tennis-ball-path inference is achieved, by combining audio and video information.

In the remainder of this chapter, we first present a video-based tennis sports analysis system for real-time implementation in Section 4.2. Then in Section 4.3, we present an audio-based scheme for detecting alternative events in automatic sports video analysis. Further in Section 4.4, we present a scheme for detecting other automatic tennis-ball-path inference for tennis sports video analysis combining audio and video information. Finally, Section 4.5 draws conclusions and summarizes this chapter.

4.2 Video-based analysis

Ideally, a powerful analysis system should be able to provide a broad range of different analysis results, rather than semantic events only, because of the various requirements from the users. For instance, object-level parameters like the real speed of players, may be helpful to certain users.

In this section, we present a fully automatic and real-time system for multi-level analysis of video sequences containing tennis sports. Our system encloses novel contributions in the following three aspects.

- An automatic *3-D camera calibration* is proposed, that enables the computation of the real-world positions of many objects from their detected image position. Such real-world coordinates are more useful for deducing the semantic events. Additionally, our calibration technique is an aid for more generic modeling of the playing field, so that it can be adapted to every court-net sports game, through only changing the court-net layout for each sport.

- An adaptively *weighted linear* model combining the visual cues in the real-world domain is proposed to identify events, since the importance of each visual cue is different for specific events. The weighting factors are adjusted adaptively for each visual cue related to different events. This ensures that our algorithm achieves a higher classification accuracy than the pure linear model.

- We build the entire framework upon the 3-D *camera calibration*, since this modeling is an efficient tool to link pixel-level analysis, object-level analysis and also scene-level analysis. Our system is capable of classifying several game events, such as service, base-line rally and net-approach events, which are consistent with the viewer's understanding about a tennis game. This framework is advanced due to its capability of providing a wide range of analysis results at different levels.

4.2.1 Overview of proposed tennis sports analysis system

Our tennis sports analysis system can be described best as composed of several interacting, but clearly separated modules. Figure 4.1 depicts our system architecture with its main functional units and the data flow.

First, playing-frame detection involves the selection of the tennis playing-field sequences out of full sports program including special scenes like e.g. breaks or advertisements. Second, court detection identifies the court location in the scenes and provides its specific information, such as size and shape. Third, the player segmentation and tracking module calculates the position

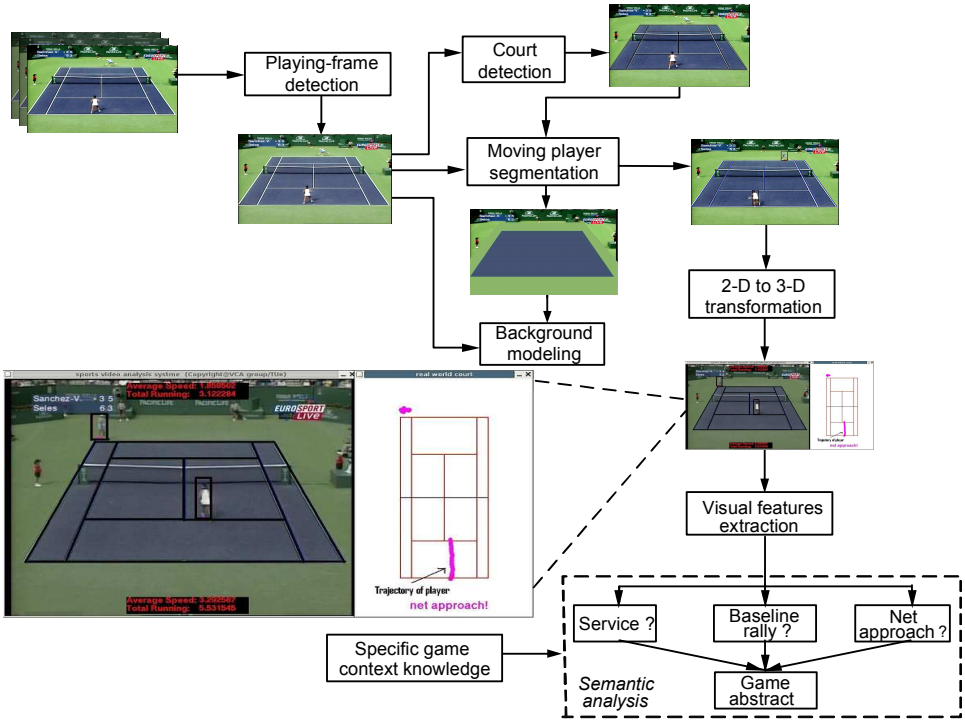

Figure 4.1: *Architecture of the complete sports video analysis system with small sample figures showing the intermediate results after each stage.*

and speed of each player, which are required to derive the player's behavior and tactics. Fourth, the camera calibration deduces a semantic meaning from the position and movements of the players, taking the camera motion into account and computes the player positions in the real-world coordinates, based on 2D-3D transformation. These coordinates are required because the player tracking algorithm only produces the player positions in image coordinates, which are physically meaningless. After the above steps are implemented, we can perform semantic analysis and classify different events, such as *service*, *base-line rally* and *net-approach*. Finally, the game abstract is obtained showing different information, such as real running distance and behavior of the players.

The most important modules involved in the analysis system are briefly explained below.

A. Playing-frame detection based on white-pixel ratio

A tennis sequence not only includes scenes in which the actual play takes place,

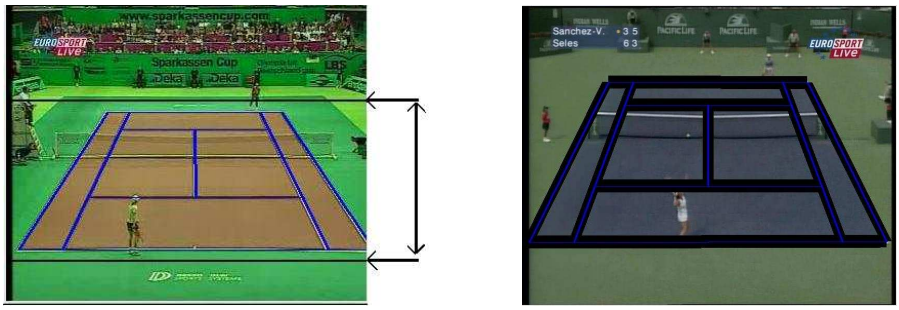

Figure 4.2: *At the left: identification of the local regions for detecting the playing field (black lines, arrows indicate the vertical for the field). At the right: the search area for the court lines (bold black lines structure).*

but also breaks or advertisements. Since only the playing frames are important for the subsequent processing, we efficiently extract the frames showing court scenes for further analysis. In our system, the playing-frame detection only identifies the white pixels of court lines and distinguishes the difference between the numbers of white pixels inside two consecutive frames. We use this metric [26], because we found that the color of the court line is always white, irrespective of the court type, and the number of white pixels composing the court lines is relatively constant over a large interval of frames (several hundreds). Compared to conventional techniques [93, 8] based on the mean value of the dominant color, this technique is more efficient and removes a complex procedure for training data.

In [93], a color-based playing-field detection algorithm is proposed that summarizes the *mean* value in each color space of the four court classes, like carpet, clay, hard and grass, based on statistical analysis using many example frames. The *Euclidean* distance between the color of the current frame and each class of courts is computed, and a class threshold is applied to decide whether it is a playing field or not. Besides a complex procedure for training data, the finding of the threshold is a serious problem, since the mean color of the same court type varies considerably, as this mean is a function of e.g. the presence of shadows, lighting conditions and partial occlusion(s).

As already mentioned, the key properties are that the color of court lines is always white and the number of white pixels in the lines is relatively constant. Based on these two properties, we propose the following playing-frame detection algorithm presented below.

1. Initialize with the court detection algorithm of [18], until a court is detected. Find the positions of two baselines and two sidelines, forming a

local area as shown in Figure 4.2 (left), where two black lines represent the vertical boundaries. The horizontal boundary is the same as the figure width.

2. Compute the fraction of the white pixels within the selected area using $F_w = N_w/N_t$, where N_w is the number of white pixels within the area and N_t is the total number of dark pixels, selected in the area. We employ the technique proposed in [18] to extract white pixels. Its advantage is that white pixels that do not belong to court lines (i.e., the player's white clothing) are rarely marked.

3. Calculate the value of $|F_w(t) - F_w(t-1)|$, where t refers to the current frame and $t-1$ to the previous frame. If it is less than a threshold, this frame is indicated as a frame containing the playing field. The threshold is experimentally determined after extensive evaluations.

B. Court detection and camera calibration

Court information, including size, shape and location, is an important aid to analyze the tennis game. To deduce the semantic meaning from the position and movements of the players, their position has to be known in real-world coordinates. However, pixel-level image processing algorithms will only calculate the player positions in image coordinates, which are physically meaningless. To transform these image coordinates to physical positions, a camera-calibration algorithm has to be applied [18]. The complete camera-calibration system comprises the following algorithmic steps.

- **Court-line pixel detection.** This step identifies the pixels that belong to court lines. Since court lines are usually white, this step is essentially a white-pixel detector. The mandatory feature of this step is that white pixels that do not belong to court lines (e.g. the player's white clothing, etc.) should not be selected.

- **Line-parameter estimation.** Once we have obtained the set of court-line pixels, we derive parametric equations for the lines. The process is as follows. We start with a RANSAC-like algorithm to detect the dominant line in the data set. The line parameters are further refined with a least-squares approximation and the white pixels along the line segment are removed from the data set. This process is repeated several times until no more relevant lines can be found.

  RANSAC is a randomized algorithm that hypothesizes a set of model parameters (in our case the line parameters) and evaluates the quality

of the parameters. After several hypotheses have been evaluated, the best one is chosen. More specifically, we hypothesize a line by randomly selecting two court-line pixels $\mathbf{p} = (p_x, p_y)$ and $\mathbf{q} = (q_x, q_y)$. For each line hypothesis, a score $s(\mathbf{g})$ is computed by

$$s(\mathbf{g}) = \sum_{(x',y')\in\Re} max(\tau - d(\mathbf{g}, x', y'), 0), \tag{4.1}$$

where $d(\mathbf{g}, x, y)$ is the distance between the pixel $(x,\ y)$ at line $\mathbf{g}$, $\Re$ is the set of court-line pixels and τ is the approximate line width. This score effectively computes the support of a line hypothesis as the number of white pixels close to the line, weighted with their distance to the line. The score and the line parameters are stored and the process is repeated with about 25 randomly generated line hypotheses. Finally, the hypothesis with the highest score is selected.

- **Model fitting.** The model fitting step determines correspondences between the four detected lines and the lines in the court model. Once these correspondences are known, the homography between real-world coordinates and the image coordinates can be computed. To this end, four intersection points of the lines $\mathbf{p}_i$ and $\mathbf{p}'_i$ are computed, and using the four resulting projection equations $\mathbf{p}'_i = H\mathbf{p}_i$, eight equations are obtained that can be stacked into an equation system to solve for the parameters of matrix H. Since the correspondences between the lines in the image and the model are not known *a-priori*, we iterate through configurations of two horizontal and two vertical lines in the image as well as in the model. For each configuration, we compute the parameter matrix H and apply some quick tests to reject impossible configurations with little computational effort. If the homography passes these tests, we compute the overall model matching error E by

$$E = \sum_{(\mathbf{p},\mathbf{q})\in\beta} min(||\hat{\mathbf{p}}', H\mathbf{p}||_2 + ||\hat{\mathbf{q}}', H\mathbf{q}||_2, e_m), \tag{4.2}$$

where β is the collection of line segments (defined by their two end-points $\mathbf{p}$, $\mathbf{q}$) in the court model and $(\hat{\mathbf{p}}', \hat{\mathbf{q}}')$ is the closest line segment in the image. The metric $||.,.||_2$ denotes the Euclidean distance between the two points, and the error for a line segment is bounded by a maximum value e_m. This bound is introduced to avoid a very high error if the input data should contain outliers introduced, e.g., by undetected lines. The transformation H that gives the minimum error E is selected as the best transformation. Note that this algorithm also works if the intersection

point itself is outside the image or if it is occluded by a player, thereby adding robustness to the system.

- **Court tracking.** When the initial position of the court is known, the computation in successive frames can be carried out more efficiently, since the position of the court in the successive frame will be close to the previous position. Tracking of the court is carried out in two steps. The first step is an initialization that captures court information by means of court-line pixel detection, line parameter estimation and model fitting [18] in the first frame containing a playing field. For each detected court line, it is simple to obtain a search area based on the start and end points of the line (see Figure 4.2). The second step executes the same detection algorithm as the first step, but now within the local search area. The current location of the court is iteratively updated to predict the search area for the coming video frame.

The above first three steps are carried out to find the initial location of the court in the first image. For the subsequent frames, only court-line pixel detection and court tracking are applied, as they are both computationally inexpensive, so that a high tracking speed is obtained.

C. Moving-player segmentation

To analyze a tennis video at semantic level, it is necessary to know where the players are positioned. Earlier systems propose several moving-player segmentation algorithms. A class of methods is based on motion detection [93, 80], in which subtraction of consecutive frames is followed by applying a threshold to extract the regions of motion. Obviously, with such a simple detection algorithm, it is impossible to analyze cases where the background is also moving, or the camera is moving at the same time. Another category proposes the use of change-detection algorithms. In change-detection systems, the background is first constructed, and subsequently, the foreground objects are found by comparing the background frame with the current video frame. The literature addressing tennis analysis [83] concentrates on selecting a video frame of the tennis court without any players as a background image and then segmenting the players in the video sequence by looking for variations within the background. Unfortunately, in most tennis videos, such frames rarely occur. In conclusion, earlier systems adopt existing techniques of moving-object detection without any exploitation of specific properties of the tennis video game, which leads to a poor detection performance. In addition, the purpose of detecting players is to obtain the player's positions. That is, only the feet positions of the players are really important for further analysis, but there is no technique addressing this feature specifically. The contribution

of our technique is also based on change detection, but we focus on building a high-quality background based on the game properties of tennis, since the performance of the change-detection technique largely depends on the quality of the background.

We have found that in most tennis video sequences, a regular frame containing the playing field mainly includes three parts: (1) the court (playing-field inside the court lines), (2) the area surrounding the court and (3) the area of the audience. Normally, the moving area of the players is limited to the field inside the court and partially the surrounding area. Moreover, the color of the court is uniform, as is also the case for the surrounding area. The above observations have been exploited to separately construct background models for the field inside the court and the surrounding area, instead of creating a complete background for the whole image. Using this concept, two advantages occur as compared to the conventional algorithms. First, a background figure with better quality is obtained, which cannot be influenced by camera motion. Second, because of the improved background picture quality, only color and spatial information are considered for further feature extraction, which makes our proposal simpler than advanced motion-estimation methods. More details about the algorithm can be found in [26].

In conclusion, the player detection and tracking algorithm is summarized as follows.

1. **Construct background for the playing field inside the court.** Up to now, we have obtained the boundaries of the court, and the coordinates of each white pixel of the court lines. We can therefore label the pixels excluding the white pixels within this area as *inside pixels*. After this, the background model for the playing field inside the court is made, in which the intensity of each pixel equals the mean intensity of all *inside pixels*.

2. **Construct background for the area surrounding the playing field.** Predict the moving area for the players outside the field, and label all pixels of this area as *outside pixels*. In order to obtain a more robust moving area, we predict it using a standard court model constructed in the real world, which is then transformed into the image domain employing 3-D camera parameters. Once all the *outside pixels* have been extracted, the same averaging technique as mentioned above is applied to construct the background for the area surrounding the playing field.

3. **Produce binary map.** Create a binary map having the same size as the original picture, and initialize all pixels with 255. This map is used later for player position analysis. A residue picture is formed by subtracting the background model from the original picture. The output

binary map is obtained by

$$B(x,y) = \begin{cases} 0, & \text{if } |d(x,y)| < T_h; \\ 255, & \text{otherwise,} \end{cases} \tag{4.3}$$

where $B(x,y)$ represents the value of the binary map at a given point (x,y), $d(x,y)$ denotes the corresponding value in the residue picture, and T_h is the threshold. Figure 4.3 shows an example, where the left figure is the original, and the right figure is the produced binary map.

4. **Extract the players**. Exploiting the knowledge of the game court, we select two *search regions* in the binary map, above and below the tennis net-line (see the right figure of Figure 4.3). Subsequently, we scan the Top and the Bottom search regions of the map (above and below the tennis net-line) for finding the player bodies. The top and bottom search region are two windows of different dimensions, in order to account for perspective differences. For searching, we locate a small player search window in the Bottom region W_B at every possible pixel. For each pixel position (the center position of the small search window), we count the number of zero values in bottom region W_B enclosed by the small window. Then we select max_B, the centroid of the bottom player, and find the related position which maximizes the zero count. Similarly, we obtain the top player position using the same algorithm.

5. **Player tracking in successive frames**. When the initial player positions are determined in the first frame with the playing field, we can detect the players in an efficient way for successive frames. We only explore two local search areas surrounding the identified small window centers of the previous frame, where the players are found. Within those two search areas, we again search for the positions that maximize the zero counts in the search areas at the bottom and top field areas W_B and W_T, respectively. In this way, the players' positions are computed at each frame.

6. **Tracking refinement**. In our framework, the semantic-level analysis requires the player position with high accuracy. It is difficult to be provided by only the player extraction and the tracking developed in Step 5. Therefore, we need a procedure that further smoothes and refines the motion of each player. Plankers [65] adopts the DES operator to track moving persons, which executes faster than the Kalman-based predictive tracking algorithm with equivalent prediction performance. Here, we adaptively adjust key parameters of the DES filter by using the real-world speed of the player calculated by our camera modeling [31]. Once the player positions in the 3-D domain are obtained with high accuracy,

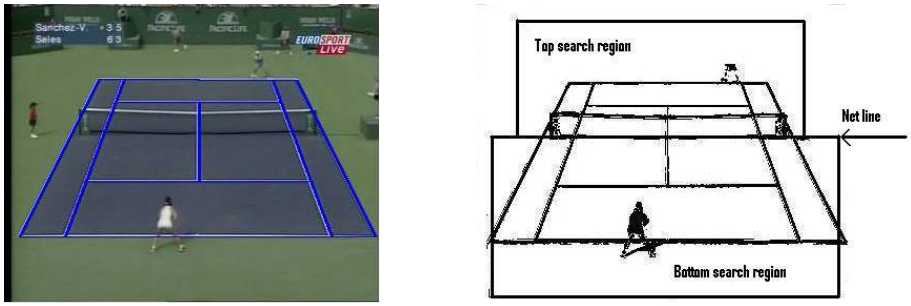

Figure 4.3: *Example of constructing a binary map with top search region and bottom region.*

the relevant parameters, such as the real speed, trajectory, and so on, can be easily computed. This kind of real-world parameters can be directly provided to the users, like a coach or the player himself. Meanwhile, the semantic-level analysis would also profit from these real-world visual features.

D. Scene-level event classification

The semantic analysis module is designed based on the definition of a set of events. Its objective is to classify various events with game context knowledge. For the user, an event would be an important mark in the play, a fault, a scoring point, etc. For the analysis, events are defined by a linear combination of a number of real-world visual cues, such as the instant speed of each player, speed change of each player, distance of the moving players to a set of reference locations (base-line, service-line) and temporal relations between each event. Moreover, we have also found that the importance of each visual cue to different events is not exactly equal. For example, the position of the player is evidently more important to identify a net-approach event, than other visual cues. Thus, we propose to assign a weighting factor to each visual cue, whose value is depending on its importance to a specific event. Such a refined weighted linear combination has the potential to yield a higher accuracy. This definition of an event has the advantage of being flexible, since any event of any time scale can be represented. Afterwards, event recognition is achieved by computing a likelihood degree, which also provides a reliability indication of the event recognition. The technical details will be discussed in the next section.

4.2.2 Semantic inference

The semantic inference derives several real-world visual cues from the image domain and afterwards, it makes weighted models for event recognition. To

achieve the first part of this task, the system should correctly bridge the gap between the numerical image features of moving players and symbolic description of the scene. To do so, we first analyze the game rules and select a list of several visual cues that really facilitate semantic analysis of the tennis game. Second, we intend to describe each key event making use of the selected visual cues from a real-world viewpoint, and further analyze which cue is more important to a specific event. The previous two steps are performed off-line, and yield mapping and computing models for events. These models are used in the algorithms for on-line computations of both steps. Third, we compute a likelihood degree of each event for each input frame and decide on the mapping of input frames to events. At the end of this step, a simple but efficient temporal filter is used to extract the start time and the end time of each event. Fourth and finally, we summarize the game based on temporal correlations among events. Let us now elaborate further for each of those principal steps.

A. Real-world visual cues in tennis video

As mentioned earlier, some existing tennis-video analysis systems [83] employ two common visual features: position and speed of the player. In this chapter, we not only extend these two cues to the real-world domain, but also propose two novel cues for event identification in tennis video, which makes it possible to detect more events. Let us now list and motivate the four real-world visual cues that are used by our algorithm. In this way, one frame is represented by the feature vector

$$\mathbf{f} = [S^I, S^C, P^R, T^R]. \tag{4.4}$$

- *Instant speed of the player S^I:* The speed of each player is definitely important, because it reveals the current status of a player (running or still) and it also indicates the intensity of the match.

- *Speed change of the player S^C:* Acceleration and deceleration of a player occurs during changes in action behavior.

- *Relative position of the player to the court field P^R:* This position is important for the recognition of those events that are characterized by a typical arrangement of players on the playing field.

- *Temporal relations among each event T^R:* In some sports games like tennis and baseball, there are strong temporal correlations among key events. For example, in a tennis video, service is always at the beginning of a playing event, while the base-line rally may interlace with net-approaches. In our case, T^R marks the first four seconds of a possible rally, starting with the service.

B. Visual-based model for each event

With the above real-world cues, we can model key events, of which three are given below. These are the events classified by our system.

- *Service:* This event normally starts at the beginning of a playing event, where two players are standing on the opposite half court, and where one is at the left part of the court, and the other is at the right part. In addition, the receiving player has limited movement during the service.
- *Base-line rally:* This occurs usually after the service, where two players are moving along their base-lines with relative smooth speeds, that is, there is no drastic speed change.
- *Net-approach:* This is one of the highlight parts of a game, in which standard visual cues: (1) a large speed change, combined with (2) close positioning of players to the net lines.

As soon as a new event starts with the playing frame of an event inside, the parameter P^R is set to unity for four seconds, after which it becomes to zero again. This marks the beginning of a sequence with an event.

All event models utilize the four real-world visual cues described earlier. We have found that a linear combination of these visual features can be applied to identify the events. Furthermore, we employ the knowledge that the importance of each cue to different events is not equal. For example, the temporal position is clearly more important than other cues in order to identify a service event, as 90% of the frames belong to the service event within the first four seconds of a playing event (verified by our sequences). Similarly, the position of the player is more important than other cues to recognize a base-line rally or a net-approach. Therefore, we assign weighting factors to each visual cue and then make linear combinations of the four visual cues. As an example, we discuss the service-event detection in more detail and illustrate the computation of a likelihood degree for an input frame. The likelihood degree L_i is obtained by

$$L_i = w_1 \times T_i^R + w_2 \times S_i^I + w_3 \times S_i^C + w_4 \times P_i^R, \tag{4.5}$$

where i is the frame number, T_i^R marks the beginning of an event, as indicated in subsection A. For service detection, if the current frame is within the first four seconds of a playing event, then $T_i^R = 1$, otherwise $T_i^R = 0$. The parameter S_i^I represents the instant speed of a player. In this case, if the speeds of both players are less than 1 m/s, then parameter $S_i^I = 1$, otherwise $S_i^I = 0$. The parameter S_i^C refers to the speed change of a player. In the service case, if the speed changes of both players are less than 1 m/s, then $S_i^C = 1$, otherwise $S_i^C = 0$. Parameter P_i^R means the relative position between the player and the court field. In the service case, if two players have positions close to the

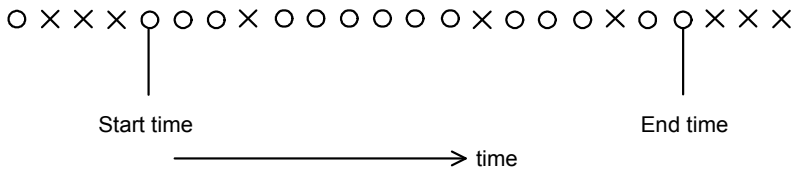

Figure 4.4: *Example showing how to extract the start time and the end time of a service event.*

baselines and also on the opposite half court, then $P_i^R = 1$, otherwise $P_i^R = 0$. Weighting factors w_1, w_2, w_3 and w_4 correspond to each feature. In the service case, $w_1 = 2$, but w_2, w_3, and w_4 are all equal to unity, as temporal relations are more important than other features. In our algorithm, when $L_i = 3$, we mark this frame as a service frame. Similarly, Equation (4.5) can also be used to extract the base-line rally and net-approach by merely changing the variable values and weighting factors related to a different event model, which makes this likelihood concept generally applicable.

C. Event extraction

Until now, each frame is classified as a service, base-line rally or net-approach. The next step is to extract the start time and the end time of each event. Figure 4.4 portrays an example, where we show a set of frames in the temporal direction. A circle represents a detected "service" frame, and a cross stands for a "non-service" frame. It can be noted from Figure 4.4 that the first frame is not a reliable start frame of the service event, although it is labeled as a "service" frame. This is because there are three "non-service" frames behind it, so that the probability that it is erroneously classified as "service" frame is large. Therefore, the first step of the start-time extraction is to measure a window for the local correlation of the i-th frame, using the following definition:

$$c(i) = s(i-2) + s(i-1) + s(i) + s(i+1) + s(i+2), \tag{4.6}$$

where i is the frame number, and $s(i)$ denotes the binary state of this frame. If frame i is a service frame, $s(i)$=1, otherwise $s(i)$ equals 0. We compute the local indication $c(i)$ for each "service" frame (circles), then select the lowest frame number i for which $c(i) > 2$ as the start frame of the service event.

In order to detect the last frame of the service, we first calculate the occupancy rate by

$$O_i = n_{j,i}/N_{j,i}. \tag{4.7}$$

Here, assuming the current i-th frame is classified as "service", $n_{j,i}$ counts the number of "service" frames between the first "service" frame with index j

and the current frame i. Furthermore, parameter $N_{j,i}$ denotes the total amount of image frames between the first "service" frame and the current frame (include some frames that are classified as non-service), hence $N_{ij} = i - j + 1$. We compute O_i for each "service" frame. The frame with the largest index i for which $O_i > 0.7$ is selected as the end frame of the service. The threshold value 0.7 was derived after conducting several experiments.

D. Game summary
Our scene-level analysis not only identifies some important events, but also intends to summarize the game, making use of time-sequential order between events. For instance, if there is a service event without a base-line rally, or a net-approach that directly changes to a "non-event", it is reasonable to deduce that such a case might be an ace or a double-fault. Furthermore, it is feasible to calculate how many net-approaches each player carried out during a match. Based on the statistical results, the player with more net-approaches is classified as more aggressive.

4.2.3 Experimental results

To evaluate the performance of our proposed algorithms, we have tested our system using 7 broadcasted tennis video clips (totally more than 40 minutes) recorded from three different tennis matches (US open, Australian open, and French open). We present various results starting with pixel-based processing, ending with event classification and performance evaluation.

A. Results for pixel and object-level algorithms
In this section, we present the results of our playing-frame detection algorithm, player segmentation and player tracking algorithm. For each of these algorithms, we compare the computed results with manually have labeled ground-truth data.

In our dataset, the system achieves a 96% detection rate on finding court-view frames among the testing frames, and 96% detection of players at the playing frames, where the criterion is that at least 70% of the body of the player is included in the detection window. Here, the ground truth data are manually labeled. Figure 4.5 portrays a set of practical visual detection results. It can be concluded from these results that our proposed algorithm not only accurately segments the player and the court, but also detects the position of the player in the image domain with different court types.

To evaluate the player position adjustment, a 70-frames clip is processed by applying the smoothing filter in the 3-D domain. Figure 4.6 shows examples of the player positions processed by various smoothing filters, where the results of our adaptive DES (see Chapter 3) compare favorably to the ground-

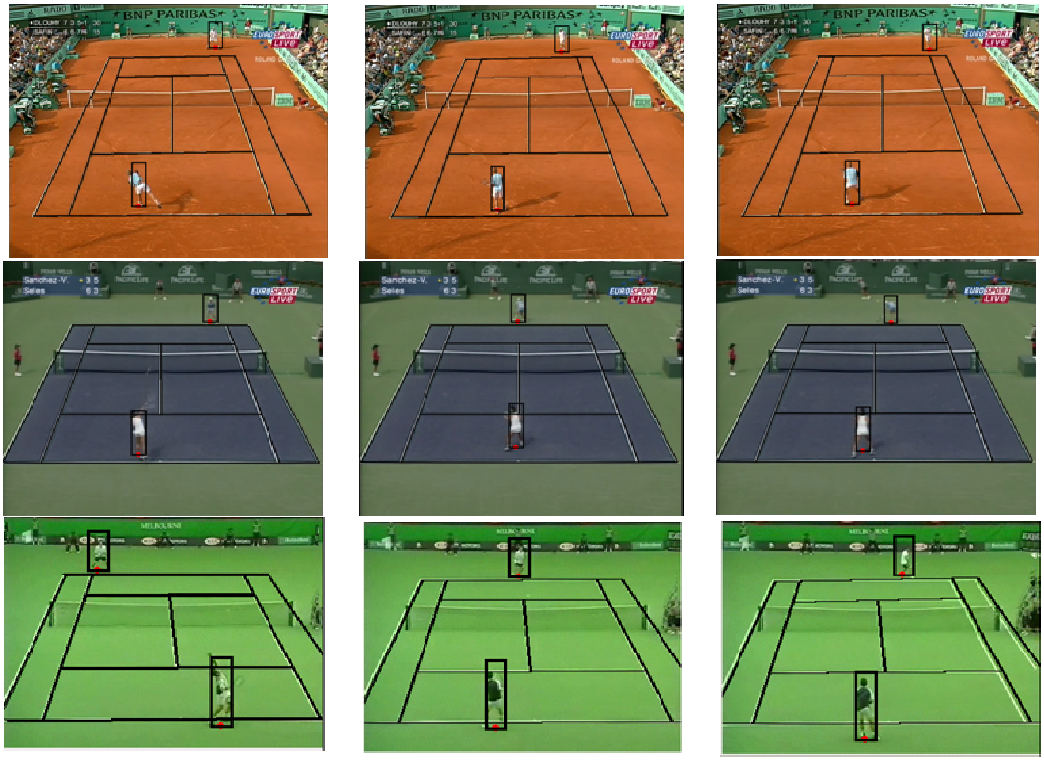

Figure 4.5: *Player and court-tracking results for 3 consecutive periods of 30 frames, where the row of sub-figures indicate tracking within that sequence (the court is indicated by black lines, the black rectangular represents the detected player).*

truth data.

B. Results for scene-level analysis algorithm
We manually have labeled all events to obtain the ground truth. At the scene level, the system automatically classifies three important events including service, baseline rally and net-approach. Table 4.1 shows the results of our simulations using the proposed weighted linear combination model and the conventional linear model. The results clearly show that our model is better than the conventional linear solution. Also, it can be concluded that the scene-level event extraction rate of the system is about 90%.

C. Results for system performance
Our video-based sports analysis system provides analysis results at three different levels, which provides sufficient analysis results for various users with

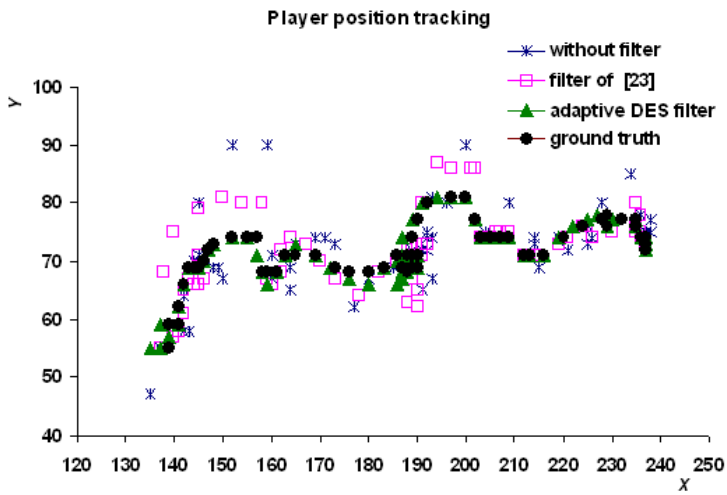

Figure 4.6: *Player position tracking, using various filtering techniques. X and Y refer to an image domain coordinate system (we track positions in the real-world domain, then transform them back to the image domain).*

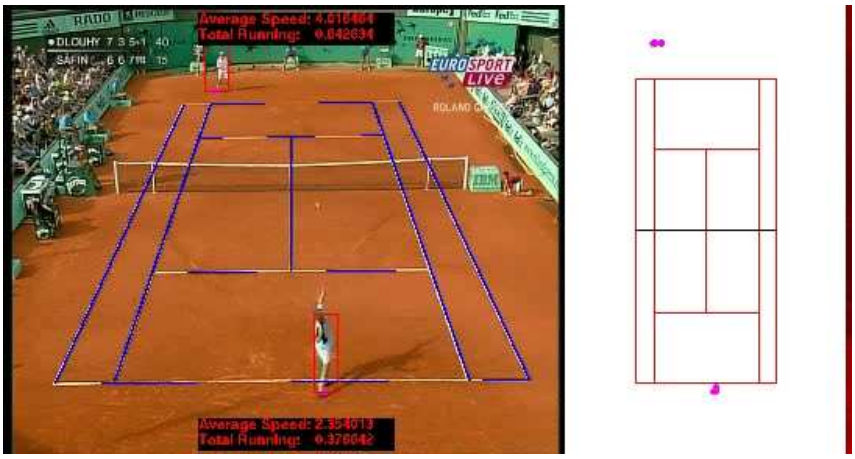

Figure 4.7: *Results of our analysis system in the service event at Frame 248.*

different preferences. An example of a full visualization for service detection is shown from Figure 4.8 through Figure 4.9. Another example of the detection of net approach is shown from Figure 4.10 through Figure 4.12. At the pixel level, several key objects are segmented and indicated. Meanwhile, the system indicates whether the current frame is a court-view frame or not. At the object level, the moving objects are tracked in the 3-D domain (at the right

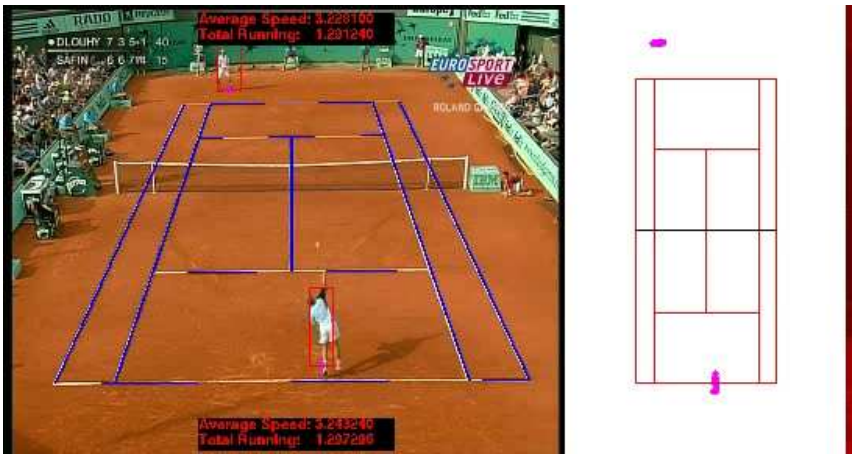

Figure 4.8: *Results of our analysis system in the service event at Frame 252.*

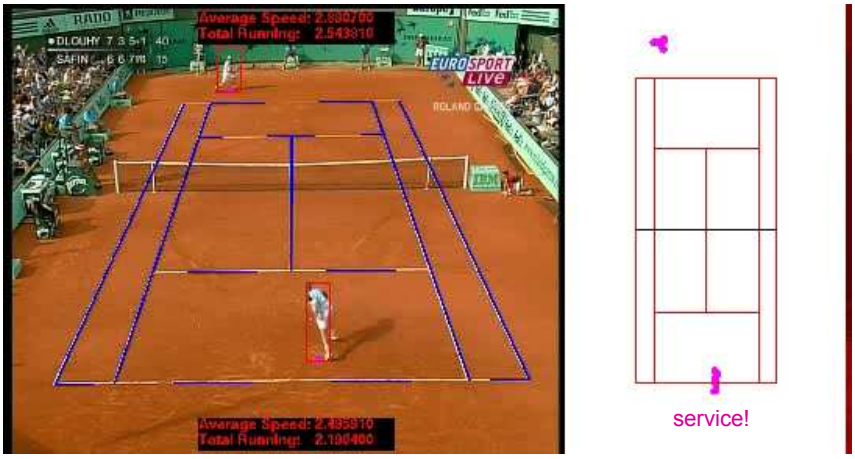

Figure 4.9: *Results of our analysis system in the service event at Frame 256.*

side of Figure 4.8-Figure 4.12). Several useful data parameters are provided, such as the instant speed of each player, the average speed of each player (in meters/second) and the total running distance. At the scene level, the system automatically classifies three important events including service, baseline rally and net-approach. We have also tested our video-based sports analyzer, achieving a near real-time performance (2-3 frames/s for 720×576 resolution, and 5-7 frames/s for 320×240 resolution, with a P4-3GHz PC).

This section has presented a nearly real-time video-based system for tennis sports analysis. It has been shown that visual cues can successfully achieve event understanding, integrating game-related domain knowledge of tennis

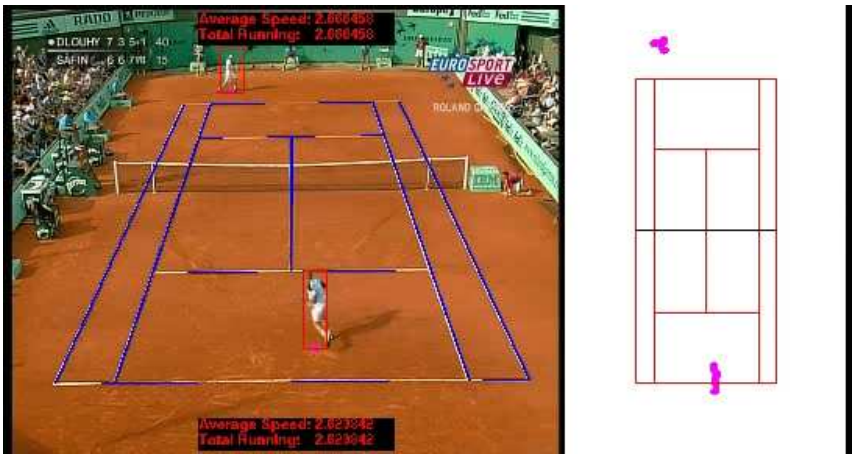

Figure 4.10: *Results of our analysis system in the net-approach event at Frame 1815.*

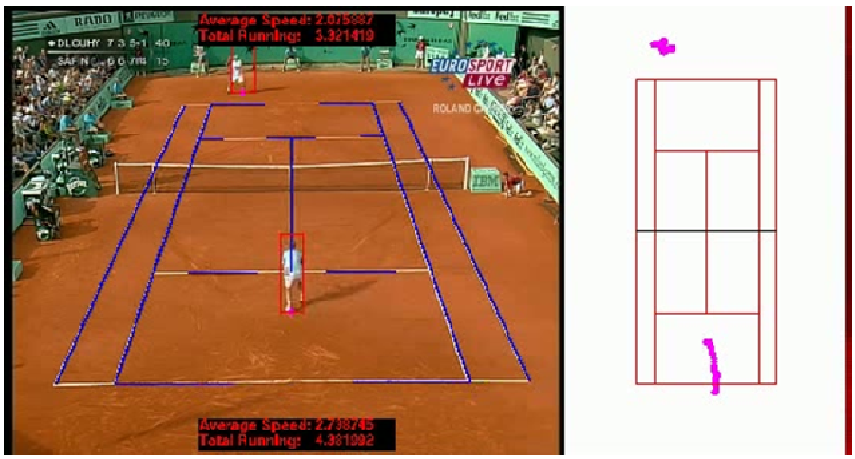

Figure 4.11: *Results of our analysis system in the net-approach event at Frame 1820.*

sports. However, it is very difficult to recognize some scenes (e.g. cheers of audience and scoring moment) only by using visual signals. Therefore, we will explore the audio signal and associate it with the video for additional scene understanding. In the next section, we will discuss this audio-based system in more detail.

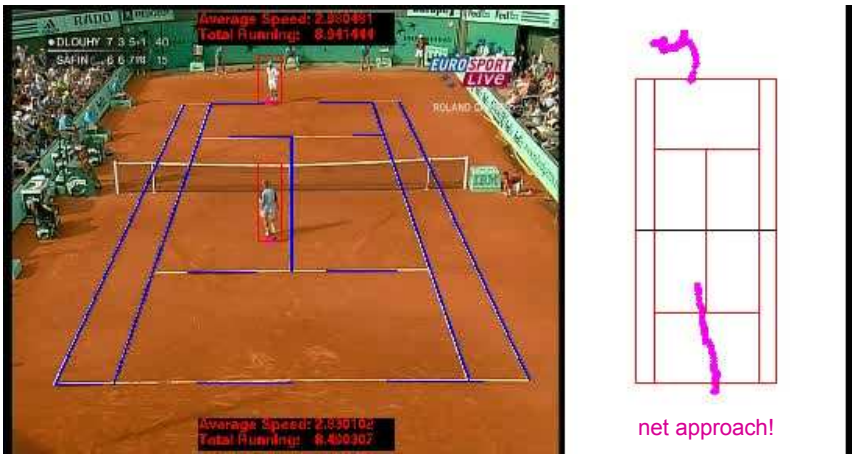

Figure 4.12: *Results of our analysis system in the service event at Frame 1831.*

Table 4.1: *Video-based classification results.*

		Total	Detected	Correct	Miss	False
Our model	Service	16	18	16	0	2
	Base-line rally	14	16	14	0	2
	Net-approach	6	6	6	0	0
Linear model	Service	16	15	14	2	1
	Base-line rally	14	13	12	2	1
	Net-approach	6	5	5	1	0

4.3 Audio-based analysis

4.3.1 Introduction of audio-based aspect

In multimedia systems, the audio signals can contribute to the semantic analysis in a similar way as with video. Figure 4.13 depicts a typical generic structure of an audio-based sports analysis system. First, audio signals are entering the *pre-processing* module, where they are divided into frames through time-windowing and amptitude normalization processing steps. Afterwards, in the *feature-selection* module, frame-level or clip-level audio features are extracted for analysis [99]. Then audio event detection is implemented in the *event-classification* module. Finally, the event information is reused in the video domain by classifying corresponding video frames into various video events.

Particular audio events like the racket-hit sound are important for analy-

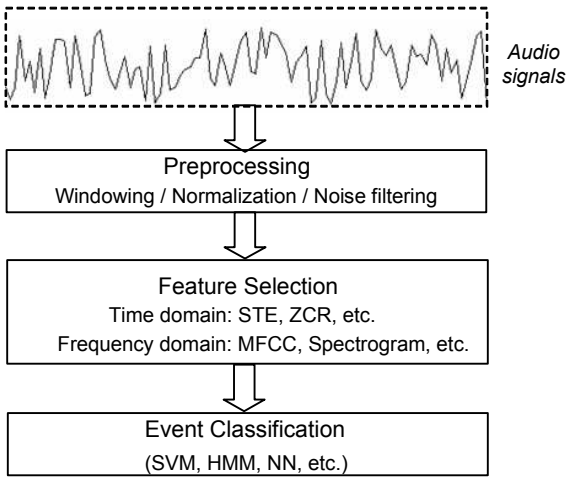

Figure 4.13: *Two modules in general audio-based sports analysis.*

sis at the semantic level. However, the short duration of the racket-hit sound and the significant ambient noise are major practical challenges in reliable racket-hit detection. The traditional approach of tennis racket-hit detection, employing template matching in the frequency domain [63] does not yield satisfactory results. Different choices of audio features [100] and learning-based algorithms [14] have been applied and may improve the detection accuracy. However, the challenge is to keep a good balance between high accuracy, robustness to varying circumstances, and the involved computational cost.

Different sports event detectors are available for the event classification module in Figure 4.13. In [100], a single-layer SVM classifier is used to evaluate the performance of a single feature. Ref. [101] uses Hidden Markov Models (HMM) for effective audio classification. These approaches may achieve good results, but they require offline processing, and additionally, a suitable dataset for training is indispensable. The objective of our research is to create an automatic online sports video analyzer of tennis video sequences at a semantic level. When using audio signals with the three-step approach from Figure 4.13, we concentrate on simplifying the second and the third module in two aspects, because they involve expensive processing. First, in the feature-selection module, we propose a three-step racket-hit detection scheme, driven by specific knowledge fused with temporal and spectral criteria of the audio event marking a racket hit. Second, in the event classification module, a simple parametric classifier using heuristic rules is constructed to implement tennis event classification without training datasets. The previous two aspects distinguish our work from the existing proposals.

The remainder of this section is arranged as follows. Section 4.3.2 intro-

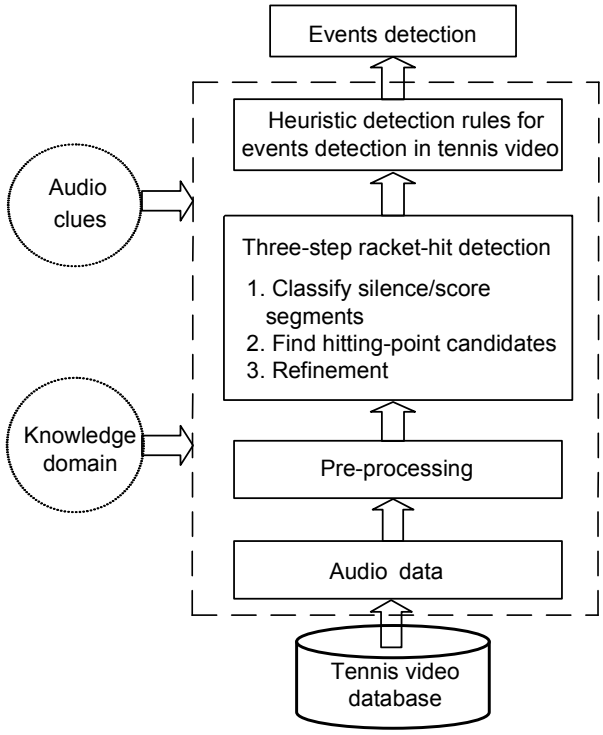

Figure 4.14: *Hierarchical block diagram of audio-based sports video analysis.*

duces the architecture of our tennis audio-based analysis system. The detailed techniques, especially the proposed three-step racket-hit detection scheme, are introduced in Section 4.3.3. Next, heuristic detection rules for audio events detection in tennis video are described in Section 4.3.4. The experimental results are demonstrated in Section 4.3.5.

4.3.2 Audio-based system framework

The hierarchical block diagram of our audio-based sports video analysis system is visualized in Figure 4.14. It serves as a supplementary customized solution for sports video analysis.

As a first step, we extract the audio data from tennis video followed by pre-processing filtering to remove accompanying noise. The audio data is used for low-level feature detection, employing both time-domain and frequency-domain analysis. After a pre-processing step, a novel three-step racket-hit detection scheme effectively removes incorrectly detected hits based on the features properties and specific game-rule knowledge. This improves the accuracy of the final event classification (see Section 4.3.3). Subsequently, in

Section 4.3.4, we employ heuristic rules for bridging the gap between feature-level space and the semantic level. Using semantic data, we are able to label various events on a tennis sports video, such as rally, service, return, audience applause/score without any visual information. In the following, the audio-based components are discussed in detail.

4.3.3 Racket-hit detection scheme

Racket-hit detection plays an important role in audio-based tennis video sports analysis. However, this task is a practical challenge as the sound of a ball impact on the racket is rather short and often mixed with significant background noise resulting from yells of players, scratches of shoes, voice of commentators and audience during the game. To achieve an accurate detection result, we propose a racket-hit detection scheme.

Step 1: Classify two segments
We initially classify a particular sequence into either *silence* or *applause/score* segments. It is a practical rule that there is less background sound during the rally, while a loud applause occurs when a player scores during the game. We use this as an essential hint for our algorithm design. The key is that the whole sequence is divided into the above two types of segments based on their different temporal audio property. After doing so, we execute the remaining racket-hit detection scheme on each segment that was classified as "silence". Then the two main processing steps follow. Figure 4.15 shows more details in feature-extraction and subsequent steps. After classifying segments as applause, two processing steps are added to find accurate hitting points in the following.

Step 2: Find hitting-point candidates
The second step of our racket-hit detection aims at searching preliminary hitting-point candidates. A racket-hit point candidate P_t over each rally period is calculated by

$$H_t = \sum_{i=1}^{n} a_i X_i, \tag{4.8}$$

where n is the number of low-level features involved, a_i and X_i indicating the weighting parameter and normalized value for a particular hitting point related to the i-th feature, respectively. Parameter H_t is the projected value from the n low-level features space for the t-th point. Hereby we set $n = 2$ as we use only two features: Short Time Energy (STE) and Spectral Power (SP). Furthermore, we set $a_1 = a_2 = 0.5$. Parameters X_1 and X_2 represent the normalized values for the two features of STE and SP, respectively. If the

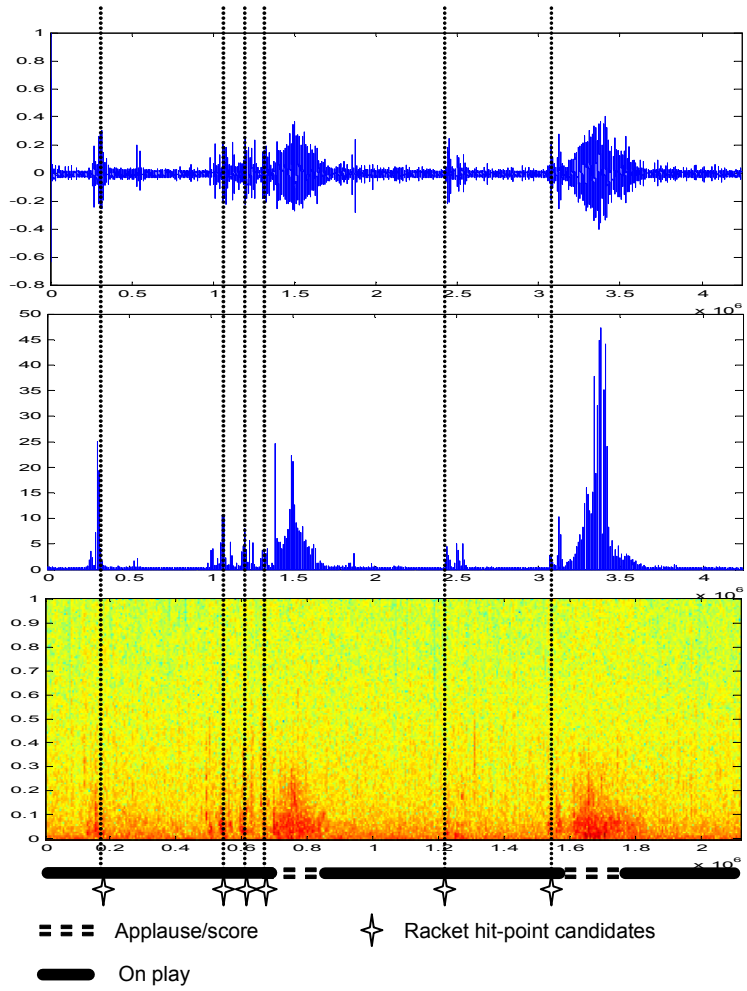

Figure 4.15: *Diagram of example result of Step 2 (above: original data; middle: Short Time Energy (STE); bottom: spectrogram).*

value H_t is above a predefined threshold T_1, we may label this point as P_t and add it in the set of racket-hit point candidates $\mathbf{P} = \{P_1, P_2, ..., P_{t-1}, P_t\}$. Evidently, Equation (4.8) may be extended when more low-level features are taken into account.

An example result of this hitting-point candidates finding is shown in Figure 4.15. The obtained segments of applause/score and rally from the initialization are supplemented with sets of preliminary hit-point candidates over each rally segment. Therefore, we proceed to Step 3 where we refine the results.

Step 3: Refinement

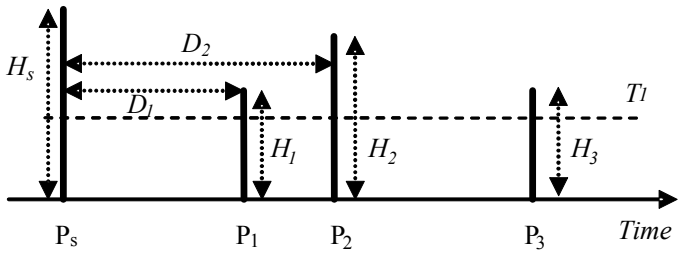

Figure 4.16: *Hierarchical block diagram of sports video analysis.*

At the third step, the refinement process corrects the detection error and thereby improves the accuracy of the final event classification.

From the length and width of a standard tennis court and the feasible speed of the tennis ball during the game, we have derived that the possible range of time between two successive racket-hits is approximately 1.2-2.0 seconds. As our video playing rate is 25 frames per second, the distance in frames between two racket-hit points is 30-50 frames. This constraint provides important information in this refinement step.

The service in a tennis game always occurs with less background noise that interferes with the racket-hit sound. Therefore, it is easier to detect than other hitting points during the rally. This is also verified by our experimental results. Based on the assumption that the service is accurately detected, we have implemented an approach to improve hitting detection, which is illustrated in Figure 4.16. Suppose a service point is denoted as P_s and its corresponding value H_s from Equation (4.8). We obtain a set of three hitting-point candidates above a predefined threshold T_1, i.e. $\mathbf{P} = \{P_1, P_2, P_3\}$ during a rally from Step 2. Their corresponding values are in the set $\mathbf{H} = \{H_1, H_2, H_3\}$. Along the time axis shown in Figure 4.16, P_s indicates the verified service point, while P_1, P_2, P_3 represent three hit-point candidates.

Subsequently, a penalty coefficient K_i is incorporated to limit the possible distance range of every two consecutive racket-hit points over a sequence:

$$K_i = \begin{cases} 1 & \text{if } 30 \leq D_i \leq 50, \\ 0 & \text{if } 20 \leq D_i \leq 30 \text{ or } 50 \leq D_i \leq 60, \\ -1 & \text{if otherwise,} \end{cases} \tag{4.9}$$

where D_i denotes the distance between the current hit-point candidate P_i and the previously verified racket-hit point, expressed in frame units. For example, to verify candidate point P_1 in Figure 4.16, the distance D_1 between the verified service point P_s and P_1 is calculated. Suppose P_1 is not verified as a racket-hit point later. We proceed to verify the next candidate point P_2 by computing the distance D_2 between P_s and P_2.

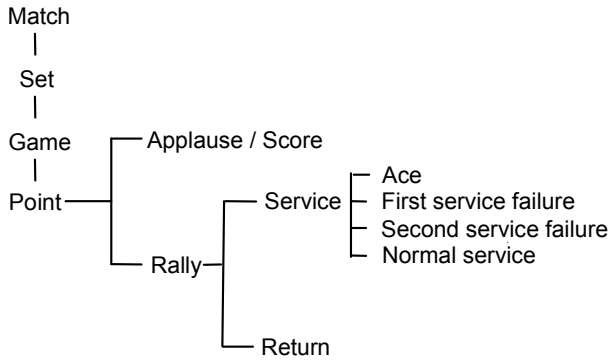

Figure 4.17: *Hierarchical semantic structure of a tennis game.*

The value of each feature-space point H_i is referred to a criterion for defining the next verified racket-hit point, specified by

$$S_i = (1 + K_i)\frac{H_i}{H_s} - \lambda, \tag{4.10}$$

where S_i is the weighting parameter for the i-th hit-point candidate P_i and λ is a threshold which is validated empirically to each candidate H_i. If $S_i > 0$, we label the i-th hit-point candidate as a validated one and add it to the set **P'**, which represents the verified racket-hit points. Then the hit-point P_i is used for the next detection iteration and calculate D_{i+1} as a reference point.

Hence, we are able to obtain the refined tennis racket-hit points **P'** resulting from Equation (4.10). Then we proceed to the next module for automatic high-level semantic sports analysis.

4.3.4 Heuristic rules for audio-based events detection

Sports videos have a game-specific structure because all sports games have particular rules and regulations. In other words, all of the games take place under a constrained environment with a defined layout. Therefore, the context knowledge of the particular game may be used as an important clue for our sports analysis. Tennis sport also applies to this general idea. To understand the relationships between various events, the particular hierarchical structure of a tennis game is shown in Figure 4.17. It forms the foundation for our heuristic rules for tennis event detection. It is also a logical verification tool for our tennis video analysis system.

Using heuristic detection rules, the tennis sports sequence can be classified into four different categories as follows.

- **Applause/Score:** a loud applause from the audience indicates that a player has scored.

Table 4.2: *Correct detection fraction of the three-step racket-hit detection scheme.*

	Recall	Precision
Racket-hit detection without refinement step	0.77	0.68
Racket-hit detection with refinement step	0.90	0.84

- **Rally:** no applause occurs. In this period, results of the racket-hit detection module further sub-divide the rally into service and return.
- **Service:** the moment that a racket-hit is detected at a silent segment. The service is further segmented into four classes.
  - *Ace:* the non-serving player fails to return in any form the opponent's service, which leads to direct loss of a point. Typically, an ace is shortly followed by an applause segment.
  - *First service failure:* a player fails to serve the ball for the first time. First service failure is characterized by a racket-hit event, which is however not followed by an applause segment.
  - *Second service failure:* a player fails to serve the ball for the second time.
  - *Normal service:* a player makes a successful service, followed by a return from his/her opponent. Practically, it is a service which is not labeled as ace, first service failure or second service failure.
- **Return:** when the opponent's racket-hit is detected after a service.

4.3.5 Experimental results

We have conducted various experiments to verify the performance of the audio-based scheme, as described in the previous subsections. Our database contains three MPEG-2 tennis video clips with a total length of 6 minutes. The audio signal is sampled at 44.l kHz sampling rate, stereo channels and 16 bits per sample.

First, we have verified the proposed three-step racket-hit detection scheme, which is described in Section 4.3.4. The three video clips contain 30 actual hitting points in total. We have defined the so-called parameters *Recall* and *Precision* of the racket-hit detection and used them to measure the performance of our algorithm. Recall is defined as

Table 4.3: *Performance evaluation of event detection in tennis videos.*

	Ground truth	Number of false detections	Number of missed detections
Score	9	0	0
Normal service	4	0	0
Ace	4	1	0
First service failure	5	0	0
Second service failure	2	0	0
Return	15	1	2

$$\text{Recall} = \frac{\text{number of correct racket hits detected}}{\text{actual number of racket hits}}, \tag{4.11}$$

while Precision is defined by

$$\text{Precision} = \frac{\text{number of correct racket hits detected}}{\text{number of all racket hits detected}}. \tag{4.12}$$

We have summarized the results in Table 4.2. It is clear that the refinement step effectively improves the recall and precision of racket-hit detection. Second, we have classified the three video clips into different events, based on the heuristic rules described in Section 4.3.4. Table 4.3 shows six mutually exclusive events and the encouraging performance of the event classification for each of them. In addition, we have constructed an audio-based tennis video analysis system. It runs under the Linux operating system and was programmed in C++. Its user interface is shown in Figure 4.18.

Let us now combine the video-based and audio-based technique to investigate the possible increase of the analysis result and discuss the details in the next section.

4.4 AV-based analysis

In multimedia systems, both audio and video signals can simultaneously contribute to the analysis at the semantic level [100, 14]. In this section, we simultaneously combine audio and video information to implement an automatic tennis-ball-path inference for tennis sports video analysis.

The tennis ball-path plays an important role in the tactical analysis of tennis. However, recovering the tennis-ball path remains a challenging task. Approaches for semantic-level analysis based on ball tracking have been explored in [63, 29, 103], where a rather accurate trajectory, using tennis-ball

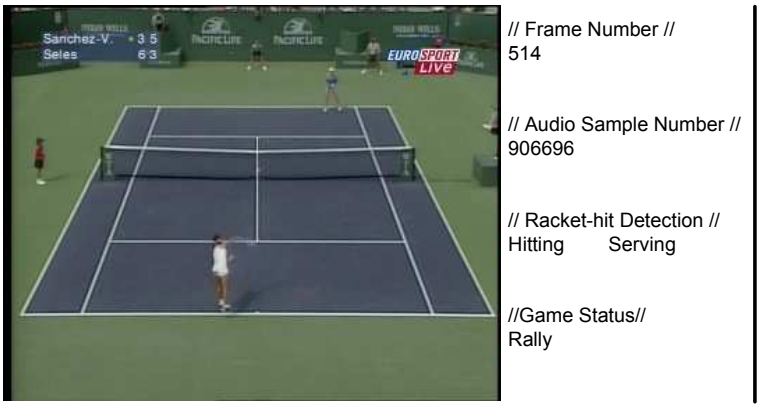

Figure 4.18: *User interface of our audio-based tennis analysis system.*

detection and tracking, was obtained. Generally, it is quite difficult to detect and track the ball accurately in practice, especially for a long sequence, because of the features of the ball in the image domain given below.

- The ball size is small.
- The detection is considerably affected by the environment, like illuminance, type of playing field, etc.
- The quality of the acquired video sequence is not always sufficient to detect the ball.
- The ball deformation is significant, especially at high speeds.

In fact, the accurate trajectory of the ball is not necessary for tactics analysis. Only the ball path is important. Therefore, we alternatively propose to utilize a non-ball-tracking approach with the fusion of audio and video signals to infer the ball path. This alternative approach should also contribute to the tactics analysis in balancing accuracy, robustness to varying circumstances, and the involved computational cost.

The objective of our research is to create an automatic online tennis sports video analyzer at high semantic level, using both auditory and visual information. This section contributes in two aspects. First, we present a simple and reliable serving-player detection scheme, that classifies the service status and which is driven by fusing audio and video information about the service. Second, the tennis-ball path is generated for tennis-game tactics analysis without detecting and tracking the ball itself.

In the sequel, we first introduce the architecture of our AV-based analysis system in Section 4.4.1. The proposed serving-player detection scheme

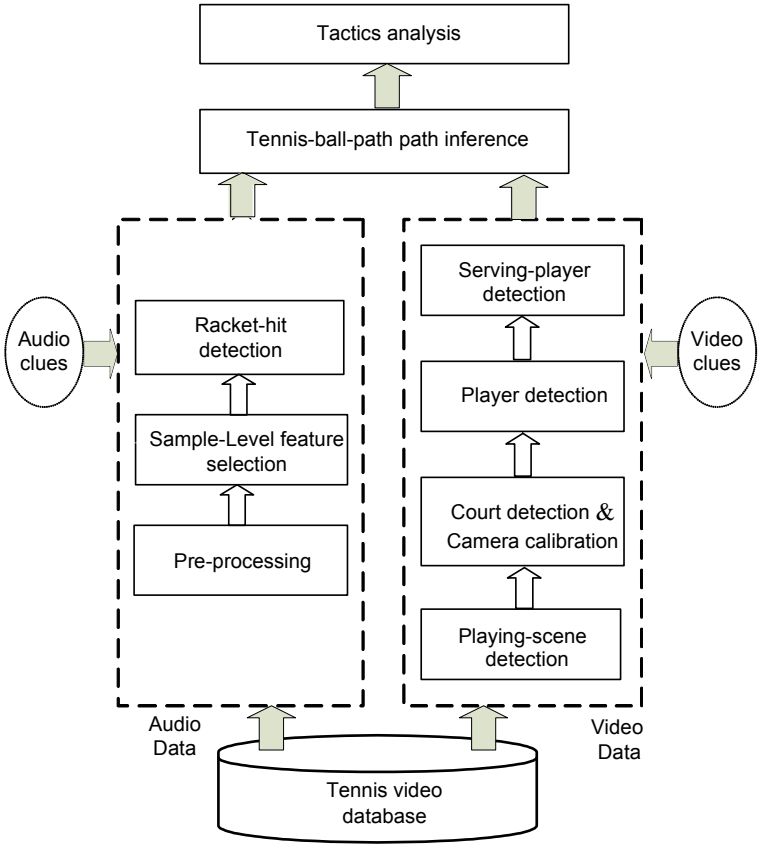

Figure 4.19: *Block diagram of our AV-based analysis system for ball-path inference and tactics analysis.*

is described in Section 4.4.2. Later, Section 4.4.3 introduces the scheme for tennis-ball-path inference. Finally, Section 4.4.4 summarizes the experimental results.

4.4.1 AV-based system framework

The hierarchical block diagram of our AV-based tennis video analysis system is visualized in Figure 4.19.

As a first step, both the video and audio data are extracted from the tennis video sequences, which are fed to two different modules. In the audio-based analysis module, pre-processing filters remove accompanying noise. Subsequently, the audio data is used for sample-level feature detection, employing both time-domain and frequency-domain analysis. Next, a racket-hit detection scheme effectively removes incorrectly detected hits, based on the feature

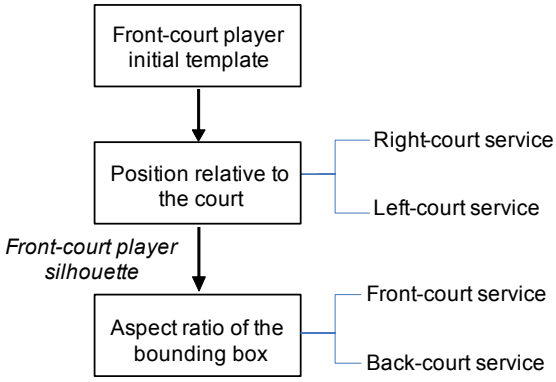

Figure 4.20: *Block diagram of analyzer for serving player.*

properties and specific game-rule knowledge [44]. In the visual-based analysis module, several techniques of our earlier work are applied to detect the playing scenes [29], track the players [29] and detect the tennis court [29]. Meanwhile, a camera-calibration algorithm [18] is used to bridge the gap between the image domain and the real-world domain. In other words, it transforms the image coordinates to physical positions. Afterwards, a serving-player detection, described in Section 4.4.2, is activated. Based on the combination of audio and video processing results, we are able to obtain the tennis-ball-path inference. Therefore, the tactics analysis of a tennis game can be implemented.

4.4.2 Visual-based serving-player detection

Triggered by the timing of each racket-hit sound, the visual analysis in the image domain is performed. Based on our previous work [29], we obtain the position of each player in each frame. Then their projection on the real-world tennis court model is also calculated. This may lead to further tactical analysis which is described in Section 4.4.3.

The serving-player detection plays an important role, as it contributes to the mapping between a sequence of racket-hit moments and the position of the hitting player in the corresponding video frames. We define a player as *front-court player* when he is closer to the camera and the other player as *back-court player*. Since the service status of two players is correlated, we analyze only the service of the front-court player. The status of the other player can be easily derived from the rules of the tennis game. For example, only one player (front-court or back-court) is serving during a tennis game and his service status is either right-court service or left-court service.

Let us now introduce our serving-player detection approach of which the block diagram is portrayed by Figure 4.20. First, the image frames corre-

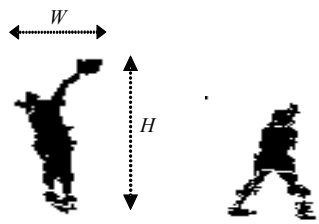

Figure 4.21: *Example of comparing the silhouettes of a serving player (left) and a non-serving player (right).*

sponding to the service event are extracted. From the results of [29], an initial template indicating the position of the front-court player, is available. As the 3-D real-world position is also calculated, we can classify the serving status into two cases, i.e. *right-court player* and *left-court player*, according to the position relative to the detected court [29]. The silhouette of the front-court player is represented with a bounding box. This enables to distinguish whether the front-court player is serving or not, based on a specific silhouette property.

Figure 4.21 shows an example of the difference between a serving posture and a non-serving posture. The feature of the aspect ratio α of the bounding box can distinguish between a serving player and a non-serving player. This aspect ratio α is defined as the ratio between width W and height H of the bounding box, which is compared to a threshold such that

$$\alpha = \frac{W}{H} = \begin{cases} \leqslant \lambda & \text{is serving, or} \\ > \lambda & \text{is NOT serving.} \end{cases} \tag{4.13}$$

Here, λ indicates an adaptive threshold which is depending on the relative court size. If the aspect ratio α of the bounding box indicates a service of the front-court player, the service status is classified as front-court service. Otherwise, the service status is labeled as back-court service. In our experiments, we have measured λ from ground-truth events and found that λ=0.8.

4.4.3 Tennis-ball-path inference for tactics analysis

Suppose we obtain a set of n racket-hit points **P'** $= \{P_s, P_1, P_2, P_3, ..., P_n\}$ from Section 4.4.2, a set of its corresponding positions for the front-court player $F = \{F_s(x_{Fs}, y_{Fs}), F_1(x_{F1}, y_{F1}), F_2(x_{F2}, y_{F2}), ..., F_n(x_{Fn}, y_{Fn})\}$, and $B = \{B_s(x_{Bs}, y_{Bs}), B_1(x_{B1}, y_{B1}), B_2(x_{B2}, y_{B2}), ..., B_n(x_{Bn}, y_{Bn})\}$, which is the back-court player. From the serving-player detection and tracking results, we are able to implement the tactics analysis, while no highly accurate ball trajectory is required.

An example of the analysis is shown in Figure 4.22. In the tennis court model, the circle and the rectangle represent the positions of the front-court

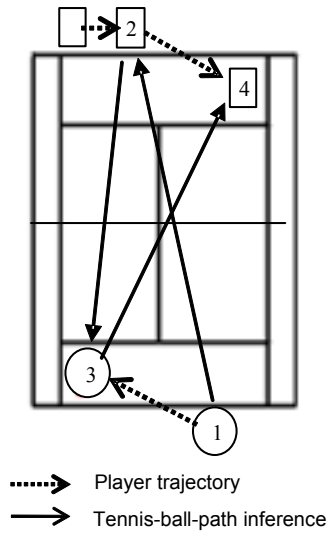

Figure 4.22: *Example of the tennis tactics analysis with indicated ball-path, playing order and player trajectories.*

player and back-court player when they are making the racket-hit, respectively. The numbers shown in the figure indicate the order of the racket hits in a particular sequence. The dotted arrow and the solid arrow represent the player trajectory and the tennis-ball-path during the game, respectively. Evidently, this arrangement contributes to the tennis tactics-analysis application. It facilitates the player and/or the coach to analyze and improve the playing performance.

4.4.4 Experimental results

Experiments are conducted to test the performance of our proposed scheme, as described in the previous sections. The database for experiments is composed of two MPEG-2 tennis video clips with a total length of 6 minutes.

First, we verify the proposed serving-player detection scheme, which is described in Section 4.4.2. The video clips contain 11 actual services in total. Given the accurate racket-hit detection results from audio-based analysis, the score of the video-based serving-player detection reaches 91% for this (limited) dataset. It should be noted that this high score relies on the audio-based analysis result. We have verified that the performance deteriorates when the audio analysis does not work properly. Employing the algorithm of Section 4.4.3, we have succeeded to classify the serving status into the four categories as shown in Figure 4.20 (front/back-court, left/right-court service).

Second, we have implemented an AV-based tactics analyzer, of which ex-

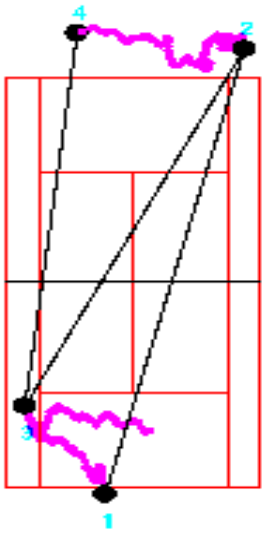

Figure 4.23: *Schematic results of tennis tactics analysis, by showing the measured ball path.*

ample analysis results are depicted in Figure 4.23. This effectively and accurately simulates a tennis game at a position level, where the tennis-ball-path is drawn in solid lines between the player positions.

4.5 Summary and conclusions

In this chapter, we have first motivated the use of a sports analysis system for consumer applications and afterwards reviewed its state-of-the-art work. Later, requirements of home-use sports analysis systems are summarized. The video-based, audio-based and AV-based detection systems are developed such that the previously mentioned requirements for home-use are satisfied.

First, we have proposed a tennis sports analysis system which is intended to be part of a larger consumer media server featuring analysis applications. The analysis system is an aid for the consumer in classifying sports video programs that are recorded in large quantities and stored on the media server. The automatic sports analysis system can generate metadata that can be used for categorizing the video scenes and give support for fast searching and retrieval. The new proposed sports video analysis system features high-level scene analysis based on real-world visual clues. The main contribution is that a selected list of real-world visual clues is applied to a set of linearly-weighted models of individual events. Robustness of event detection is achieved by using time-line models, in which the player actions become sequantially visible. This

produces probabilistic values indicating the reliability of the event occurrence. Furthermore, the high-level sub-event extraction rate of this system is about 90%. The complete proposed application is efficient enough to obtain a real-time or near real-time performance (2-3 frames/second for 720×576 resolution, and 5-7 frames/second for 320×240 resolution, with a P-IV PC running at 3 GHz). The system may be extended to analyze other sport types [32], like volleyball, badminton and basketball, since several techniques, such as court detection, camera calibration, player segmentation and high-level analysis, are generally applicable to other sports.

In the second part of this chapter, we have presented a scheme for automatic sports video analysis, based on audio signals only and specific game context knowledge. A three-step racket-hit detection algorithm is employed to achieve accurate event classification. The algorithm applied a linear combination of short-time energy and spectral power, followed by a refinement step. Besides this, heuristic rules, based on specific knowledge of the tennis game, are used for mapping the sample-level features to the semantic level. It was shown that the system implementation can identify meaningful events such as rally, scoring, different types of service, and return. The experimental results indicate that we have achieved a detection accuracy up to about 90% for racket-hit detection. It is very difficult to recognize some scenes (e.g. cheers of audience and scoring moment) only by using visual signals. Therefore, we have explored the audio signal and associate it with the video for additional scene understanding.

In the third part of this chapter, we have discussed a scheme for automatic tennis-ball-path inference for tennis sports video analysis, based on simultaneously combining audio and video information. First, an effective serving-player detection algorithm is employed which aims at locating each player in the image domain, corresponding to each racket-hit that is detected by audio-based analysis. The player detection and tracking is an aid in finding the service by the aspect ratio of the player silhouette and afterwards accurately classifying the service into several categories such as left-court/right-court service and front-court/back-court service. Second, the tennis-ball-path is generated for tennis game tactics analysis without detecting and tracking the ball itself, by projecting players in a real-world model and combining previous analysis with game rules. It was shown that our scheme can deliver a tactical analysis based on the fusion of video and audio analysis modules. The player/tactics analysis is improved by showing the ball path.

This chapter has discussed a study case how to integrate the domain knowledge in event-based understanding and employ 3-D camera calibration to perform semantic analysis of the scene. In order to meet the *real-time* requirement for embedded applications, fast and effective algorithms are selected and implemented at each processing step. We can also extend the AV-based analysis system for robust event understanding by employing audio

information. Audio signals can be utilized to recognize cheers of audience and scoring moments, which are unlikely detected by using only visual signals. Therefore, it is interesting and useful to include audio-analysis functions into consumer/embedded content-analysis systems for enhanced robustness. More importantly, the techniques utilized for visual analysis of tennis sports can also be applied to other embedded applications, such as surveillance. For example, a gun-shot detection [79] significantly attributes to a robust detection of bank robbery. Evidently, new specific domain knowledge (e.g. the definition of abnormal event and environment configuration) is required for system design and algorithm selection. We will discuss the associated technical details for surveillance applications in Chapter 5.

Chapter 5

Event understanding in a surveillance application

5.1 Introduction

5.1.1 Motivation

The previous chapters have discussed the fundamental techniques for human motion analysis and have applied them in a specific application: sports video analysis (Chapter 4). In this application, scene understanding is achieved by tracking the players in the field and interpreting their playing actions. In this chapter, we will apply person detection to different application from the surveillance area. Moreover, we also study and present an experimental architecture featuring software components and close to real-time execution.

Video surveillance can contribute to the safety of people in the home or other places and also facilitate and ease control of home-entrance and public areas. A key function in a surveillance system is the understanding of human behavior. The automated analysis of human behavior in surveillance applications is the subject of this chapter, with the aim to explore efficient algorithms for safety use. As a consumer video application, automatic surveillance requires a sufficiently high accuracy and the computation complexity should enable a real-time performance. For such a system, we need to analyze not only the motion of people, but also the *posture* of the person, as the postures of the persons can provide important clues for the understanding of their activities. Hence, accurate detection and recognition of various human

postures contribute to the scene understanding. Furthermore, there is a relentless pursuit of more effective and efficient management of human-motion analysis results, which enables quick retrieval of video sequences, containing important events, such as a burglary and/or falling incidents of elderly people. For example, a video database can be used to search for important events and classify those events, based on the semantic analysis of the human motion and their interaction behavior in the scene.

5.1.2 Related work in surveillance systems

Most real-time surveillance systems have focused on understanding the events through the study of trajectories and positions of persons, using *a-priori* knowledge about the scene. Such systems [11, 33] can monitor activities over various scenarios, using single or multiple cameras. They can detect and track multiple persons and vehicles within cluttered scenes and manage their activities over a long period of time. However, the monitoring performances of the above systems mainly rely on the detected trajectories of the concerned objects. The results are not sufficient for event analysis in some cases. As the local properties of the detected persons are missing, the developed systems lack the semantic recognition result of dynamic human activities. Some visual systems for indoor human activity monitoring [108, 75] track humans and classify postures in a home-living environment. Furthermore, the above systems extract important activity statistics and functional assessment data from videos in a hierarchical structure. However, no execution-time performance is reported from [108, 75] and we assume that they cannot achieve real-time performance. Therefore, it is important to find efficient algorithm solutions while keeping the high analysis accuracy. Also, the *combination* of using trajectory, posture recognition, and camera communication is required to improve the semantic analysis of the human behavior.

Multiple cameras are utilized in surveillance systems to improve detection accuracy. Different cameras are connected as a network and camera calibration lays a solid ground for information fusion from different viewpoints. However, most multi-camera setups [75, 9] rely on manual camera calibration and they always require initialization when camera locations or capturing environments change. Therefore, it is necessary to design an automatic *camera calibration* scheme to extend the feasibility and flexibility of a surveillance system. Furthermore, camera calibration techniques facilitate scene reconstruction in 3-D space, which plays a useful role in semantic-event analysis of multimedia applications [27]. The accurate and realistic reconstruction in a virtual space can significantly contribute to the scene understanding, like crime-evidence collection and tactical analysis. Therefore, it is interesting to extend scene-reconstruction functionality in advanced surveillance applications, such as home-care monitoring and robbery-detection surveillance. The

3-D scene reconstruction can be conducted to visualize the scene for further analysis. In the application of a bank-robbery detection, e.g., this extended processing is useful in the crime-scene analysis, data retrieval and evidence collection.

From the literature study, we find that it is important to design efficient algorithm solutions, while keeping the high analysis accuracy. Also, we should propose a behavior analysis system which can provide a combination of trajectory, posture recognition, and event recognition, ranging from pixel-based to event-based analysis. Furthermore, the system should be extensible for using multiple cameras, which is necessary to address the occlusion problem. Also, an effective scheme to perform camera communication is required.

5.1.3 Requirements of surveillance analysis systems

Surveillance analysis in embedded applications offers various interests for the users of the system. The specific challenges for consumer applications are as follows.

1. Scalable and accurate results should be provided for an embedded surveillance application.

2. A description of behavior and semantic events should be provided and transmitted over the network in order to exchange information between the processing components of a large and distributed surveillance system.

In the sequel, we will discuss the above requirements in more detail.

- To address the challenging problem of accurately analyzing human motion and achieving high-level event analysis from monocular or multi-view video sequences, the system should provide analysis at different semantic levels. A joint analysis tool is required to bridge the gaps between the pixel-level, object-level and event-level analysis and classifications. For example, the trajectory can indicate whether the person enters the restricted area in a scene. The individual posture can also specify the individual action, and analyze interaction modeling. Our system has been designed such that it incorporates multiple levels of human motion and posture analysis from the object level onwards. The system can be utilized in surveillance applications with analysis results at four levels of processing, which will be addressed later in this chapter.

- With respect to the second requirement, we have organized an evaluation of our framework by partly embedding it in a new experimental real-time AV content-analysis system. This setup was developed in cooperation with the industry in the framework of the European ITEA

project Cantata. In that framework, the ViPER file format is applied to provide an effective description of the information which communicates over the network. The evaluation of this framework at the end of the project has results in an experimental system achieving a near real-time performance (13-15 frames/second). Further details will be presented in Section 5.3.

To accurately analyze human motion and rapidly detect abnormal events at a high semantic level, we contribute in three aspects in this chapter.

- *Component-based architecture.* First, human behavior analysis algorithms are directly designed for real-time operation and embedded in an experimental run-time AV content-analysis architecture. The run-time architecture is designed to be generic for multiple streaming applications with a *component-based* architecture. It facilitates advanced video content analysis for surveillance, featuring network-based communication.

- *Hierarchical human motion.* Second, a flexible framework is proposed to enable hierarchical human motion analysis. It can be utilized in surveillance applications with four-level analysis results, using single or multiple cameras. The motion analysis at higher levels contributes to object and event understanding.

- *3-D reconstruction.* Third, a 3-D reconstruction scheme is introduced for scene understanding, based on automatic camera calibration. The location and posture of persons are visualized in a 3-D space after context knowledge is integrated. More specifically, the 2D-3D mapping provides a platform for normalized motion configuration (i.e. location and speed) and scene visualization in the real world.

A summary of frequently used terms and their definitions, which are presented in this chapter, is shown in Table 5.1.

In the sequel, we first introduce our system design, based on this software architecture presented in Section 5.2. Afterwards, Section 5.3 presents the overview of a run-time surveillance analysis system. As a key component embedded in the run-time system, the human behavior analysis framework is introduced later in Section 5.4. Then, Section 5.5 describes the techniques applied for the behavior analysis framework. The experimental results on surveillance video are provided in Section 5.6. Finally, Section 5.7 concludes this chapter.

Table 5.1: *Frequently used terms and definitions*

Level	Term	Definition
Feature-based	State	Spatio-temporal signal property valid at a given instant or during a time interval. A state characterizes only one or more features of a moving object or a moving object with relations to other physical objects.
Object-based	Trajectory	Sequences of temporal locations of an object, that is extracted by visual tracking. A path reflects the trajectory of a moving object in 2-D or 3-D space.
	Action	Resulting state of a moving object, that is conducting a task (e.g. human posture indicating pointing to a person).
	Activity	Specific action performed by a subject (human in our investigated case).
Event-based	Event	The occurrence of an activity or some activities in a particular place during a specific time interval. It is characterized by two attributes: its spatial location (positions of the moving objects involved in the scene) and its temporal relationship (during a certain interval).
	Interaction	Event that occurs as two or more objects have a mutual influence on each other.
	Behavior	High description of human activities of one or more persons, e.g. meeting, discussing, where the context of the behavior indicates a certain security impact (threating or not).
	Scenario	Predefined sequence of events.

5.2 System design based on software architecture

Software architecture deals with the design and implementation of the high-level structure of the software. It results in assembling a certain number of architectural components in some delicately chosen forms to satisfy specific functionality and performance requirements of the system. To design a real-time behavior analysis system and implement it into an architecture, we have employed a well-known model of software architecting: the "4+1" view [77]. This model (see Figure 5.1) consists of five main views:

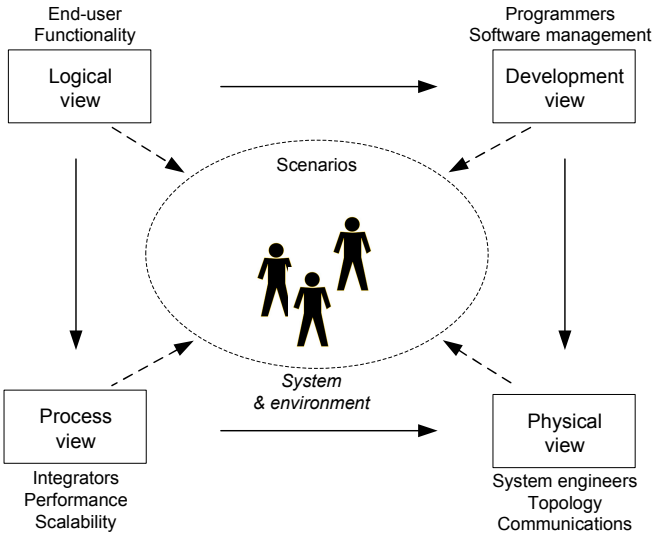

Figure 5.1: *"4+1" view model of architecture, after [77].*

- The *logical* view, which describes the (object-oriented) system in terms of abstractions, such as classes and objects. The logical view typically contains class diagrams, sequence diagrams, and collaboration diagrams. Other types of diagrams can be used when applicable.
- The *development* view, which describes the structure of modules, files, and/or packages in the system. The package diagram can be used to describe this view.
- The *process* view, addressing the processes of the system and how they communicate with each other.
- The *physical* view, which describes how the system is installed and how it executes in a network of computers. Deployment diagrams are often used to describe this view.
- The *use case* view, which provides a scenario for the functional aspects of the complete system. This view can be presented using case diagrams and use-case specifications.

In the following, let us explain every view and associate it with our investigated cases. Some examples are provided to explain individual views in our architecture design, when the "4+1" view model is applied.

A. Logical view
The logical view supports behavioral requirements and shows how the system is decomposed into a set of abstractions. Classes and objects are the main elements studied in this view. We can use class diagrams, collaboration diagrams and sequence diagrams to show the relationship between these elements from a logical view. For example, class diagrams show classes and their attributes, methods, and associations to other classes in the system.

The class diagram hardly provides a complete profile of the system for two reasons. First, class diagrams are static, so they cannot indicate how the system will react to user input. Second, class diagrams are often too detailed to offer a useful overview of the system. Collaboration diagrams (or communication diagrams) and sequence diagrams are used to see how objects interact in the system. A collaboration diagram is a simple way to show system objects and the messages that pass between objects. Collaboration diagrams are very practical for showing a birds-eye view of collaborating objects in the system. If a more detailed window into the system's logic is required, drawing of a sequence diagram is a good option. Sequence diagrams provide more detailed information than collaboration diagrams. Therefore, architects and designers often use sequence diagrams to fine-tune system design.

B. Development view
The development view is used to describe the modules of the system. Modules are bigger building blocks than classes and objects and vary according to the development environment. Packages, subsystems, and class libraries are all considered modules. The development view can be used to study the storage locations of actual files in the system and development environment. Alternately, it is a straightforward way to view the layers of a system in a layered architecture. A typical layered architecture may contain a User Interface layer, a Presentation layer, an Application Logic layer, a Business Logic layer, and a Persistence layer.

C. Process view
The process view describes and studies the system processes and presents how they communicate with each other. An overview of the processes and their communication contributes to averting unintentional errors. This view is useful when multiple and simultaneous processes or threads exist in a software.

The process view can be described from several levels of abstraction, starting from independently executing logical networks of communicating programs. The process view takes into account many of the non-functional requirements or quality requirements like performance, availability, etc. Activity diagrams are typically used to describe this view.

D. Physical view
The physical view discusses the mapping(s) of the software onto the hardware and reflects its distributed aspect. Specifically, this view provides how the system setup is installed and how it executes in a computer network. This view takes non-functional requirements (e.g. availability, reliability, performance, and scalability) into account. For example, the installation of a camera network and the setup of the calibration grid in our surveillance application is carefully considered in the physical view.

E. Extra view
The "plus-one" view takes alternative view on the architecture, which is specific for the study application. For example, in many systems, the real-time performance of the architecture is essential in such system, a specific performance analysis of the system is performed. Similarly, extra views can be studied with respect to complexity, cost, etc.

In the "4+1" view model, use cases are employed to explain the functionality and structures described by other views. The use-case view consists of use-case diagrams and specifications, detailing the actions and conditions inside each use case. Later in our experiments, several scenarios (home-care monitoring and bank-robbery detection) are tested to show the adaptiveness of the architecture design.

After the "4+1" view model for software architecture is considered, we proceed to design a networked execution system for surveillance applications, which will be discussed in the next section.

5.3 Networked execution system for surveillance applications

5.3.1 Overview of component-based framework

Surveillance analysis in consumer applications offers various ways for interacting with the surveillance data for the user of the system. A specific challenge for consumer applications is that it requires high-processing efficiency achieving (near) real-time operation with low-cost consumer hardware is highly required.

For this purpose, we have organized an evaluation of our framework by partly (in this chapter, only the single-view case in robbery detection) embedding it in a new experimental real-time AV content-analysis system, as developed in the European project Cantata, which is illustrated in Figure 5.2. The Cantata framework is aiming at real-time performance in surveillance applications. To this end, a special execution architecture is designed. More

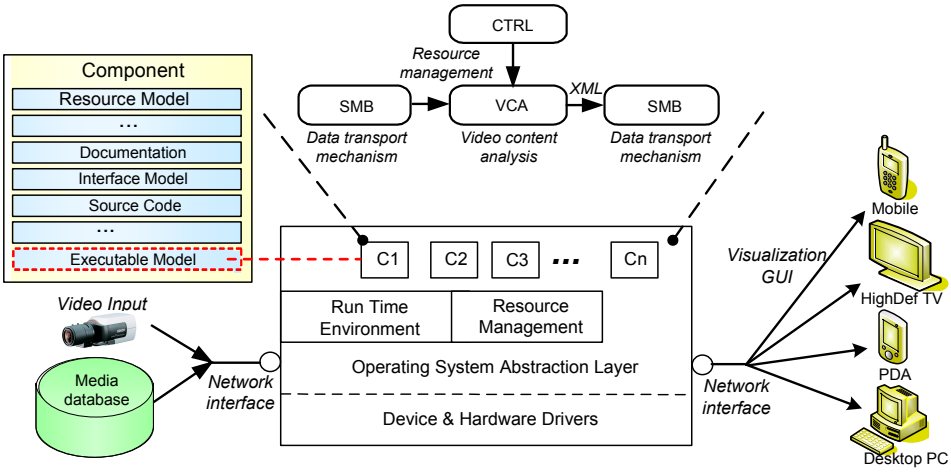

Figure 5.2: *Block diagram of the Cantata framework.*

specifically, this run-time system should be generic for multiple streaming applications with component-based architectures. The system facilitates advanced video content analysis (a surveillance application in our case), connecting via a network. The surveillance video files, which are captured online from an advanced embedded camera or is retrieved from a media database, are streamed to our analysis component (VCA block, Figure 5.2). Afterwards, the resulting XML data from the analysis component is streamed to the real-time visualization GUI terminals (PDA, mobile, PC, etc.) through the Streaming Memory Buffer (SMB). Besides this, resource management is applied to optimize the content-analysis processing, even in resource-constrained conditions (computation power, memory, etc.), i.e., when not all resource requirements of all components can be satisfied at the same time.

In a component-based environment, an application is a collection of components which are connected via their interfaces. By connecting them, the functionality of all separate components is combined to an application. The most important role of the Run Time Environment (RTE) is to build applications by loading the separate components (classes), thereby making instances of the loaded components, and connecting these instances (objects) via the interfaces. The structure and configuration of the application is modeled in an XML file. This file is the input for the RTE when creating the application. There are two cases of creating the application supported by the RTE.

In the first case, the components are dynamically loaded into memory. When components communicate with each other, interfaces of these components must be connected at run-time. Connecting an interface requires that pointers to the interface functions inside the component should be available.

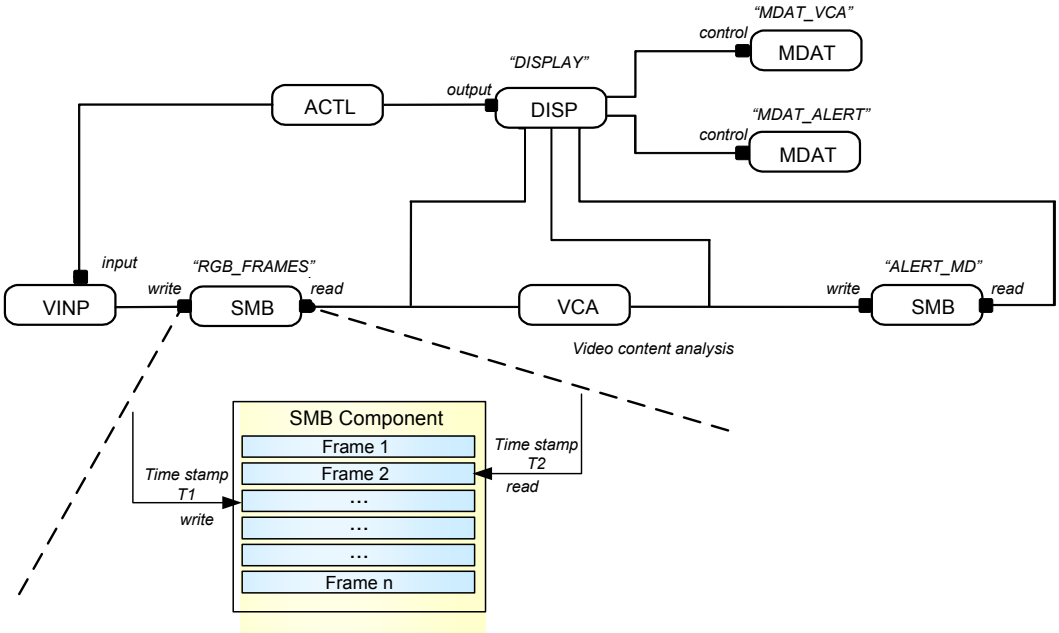

Figure 5.3: *Block diagram of the component-based data flow. The SMB component is shown in detail.*

This property requires support of the Operating System (OS). In the second case, there is only limited OS functionality or even no OS at all. This case is typical for embedded systems implementation. The components and interfaces are linked statically to one executable module.

In Figure 5.2, the Resource Management Component (RMC) is a supplementary part of the Cantata Runtime Environment. The RMC is responsible for managing the memory for the components. It provides interfaces to request and release memory budgets, and to allocate and free memory within the budgets. The RMC guarantees memory access by negotiating memory budgets with components. Upon a budget request, the RMC checks and accommodates for sufficient space, within the memory space managed by the RMC. If so, then the budget is granted. Afterwards, the memory is allocated within the budget for each component within the framework.

5.3.2 Specific components involved in the data flow

The component-based data flow is illustrated in Figure 5.3. Each component within the data flow should describe its interfaces, which are employed by the RTE to determine the functions and perform the tasks. Each component within the component-based data flow is briefly introduced below.

- *SMB - Streaming Memory Buffer*
  Two streaming components are typically connected via a buffer mechanism to pass data from one component to the next in the data-flow graph. A component that produces output data requires memory chunks in the buffer to write the output data. The consuming component obtains a reference to these memory chunks containing the data written by the producing component. These references are used to read the data from the buffer. The data are read in the same sequence as they are written. The buffer can be seen as a passive streaming component. Execution of the functionality is performed on the thread of execution of the calling component. The SMB component is used to connect components with each other. It allows data to be written and read chronologically.

- *DISP - Display Alert Component*
  This component visualizes the alert that is generated by the semantic-based alert generation component. Furthermore, the bounding boxes are generated by the 3-D behavior analysis component and superimposed on the live decompressed video on the output display. This will combine the output of the decompression component, the output of the 3-D behavior analysis component and the alert generation component. DISP can show two windows. The first window presents the input video, overlaid with metadata information from bounding boxes around persons, texts about posture characteristics, reliability figures and alert messages. Optionally, the second window shows the floor map of the scenario containing objects like a counter and restricted areas in a bank hall, where also ground location coordinates of identified persons are rendered. DISP is flexible in accepting metadata from the VCA component, as DISP can be instantiated multiple times and share the GUI resources.

- *VCA - Human behavior modeling*
  This component performs the human behavior modeling and provides the alert signal when abnormal event is detected. The technical detail will be presented in Section 5.4. It consists of three main functions: First, at the trajectory-based level, every moving person in the scene is detected and tracked. The position of every person is indicated in the 2-D image domain and the 3-D world domain. An automatic camera-calibration algorithm is implemented to facilitate the transform between 2-D position and 3-D position. Second, at the body-based level, the posture of each person (e.g. pointing, squatting, raising hands above, etc.) is classified and analyzed. Third, in the event-based analysis, this component aims to detect, in (nearly) real-time, abnormal or risky situations using a semantic representation of the context (e.g. a bank robbery) and logic rules describing such situations. Various alert levels associated with corresponding information, such as the situation, people

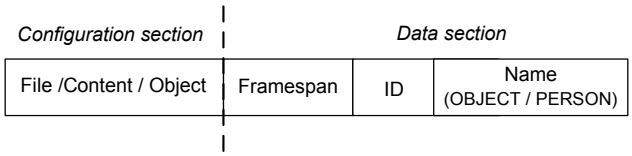

Figure 5.4: *Basic structure of the ViPER file format.*

and objects involved, will be generated. The above results are represented in the ViPER xml format and passed to other components in the run-time environment.

- *ACTL - Application Controller*
  This component accepts user input and controls other components. The Application Controller will tell the other components to either start or stop. A user can tell the application to stop the whole process and terminate. ACTL cannot interrupt the sequence once it has started.

- *VINP - Video Input*
  This component reads a file with recorded video, and passes control information and video frames. VINP is used during the development of the RTE platform. Each video frame is passed to the VCA component through the SMB component.

- *MDAT - Metadata utility component*
  This component parses a metadata stream, stores metadata-based information, and returns information to retrieving components. The MDAT component is able to parse a streamed XML sequence. The VCA component passes this XML sequence on to the MDAT through the DISP component.

We utilize the ViPER file format to represent XML data within the framework. The ViPER format is often used at the VCA component as a suite of tools. This suite was originally designed for the evaluation of video analysis, such as tracking people, detecting text, etc. The original ViPER file format does not provide a special description, when the format is used in a streaming environment. The structure of the ViPER file is described in Figure 5.4. The ViPER file format is an XML format, the root element of its XML structure is <viper>. There are two sections in this element, <config> and <data>. The configuration section defines the descriptors and the data section instantiates descriptors for recognized objects or persons. For more detail, we refer to Appendix B in the thesis.

Some problems occur because a streaming file cannot provide the actual timeline, when real-time performance is required. For example, the end time

Table 5.2: *Example of XML file for robbery detection.*

```
<?xml version="1.0" encoding="UTF-8"?>
<viper>
    <config>
      <descriptor name="ALERT" type="CONTENT">
        <attribute name="ALERT_LEVEL" type="dvalue" />
        <attribute name="LABEL" type="svalue" />
        <attribute name="DESCRIPTION" type="svalue" />
        <attribute name="INVOLVED_OBJECT" type="svalue"/>
      </descriptor>
    </config>
    <data>
      <sourcefile filename="alert.mpg">
      <content timespan="154245:156483" id="Alert1" name="ALERT">
        <attribute name="ALERT_LEVEL">
          <data:dvalue value="1" />
        </attribute>
        <attribute name="LABEL">
          <data:svalue value="RobberyDetected" />
        </attribute>
        <attribute name="DESCRIPTION">
          <data:svalue value="A robbery has been detected." />
        </attribute>
        <attribute name="INVOLVED_OBJECT">
          <data:svalue value="125" />
        </attribute>
      </content>
      ...
      </sourcefile>
    </data>
</viper>
```

of the meta data is not known on the first appearance of the meta data. Furthermore, if circular storage is used for security reason, the static file storage cannot be used as the meta data file continuously changes. To address the above problems, we modify the original ViPER file format. The frame-span attribute of each object is replaced by a time-stamp attribute. Meanwhile, to avoid data redundancy, the VCA component should send new or updated data (the modified part) on objects only when their states change. Here we show an example of reference XML data (in the ViPER file format) for alert generation when a robbery is detected in Table 5.2.

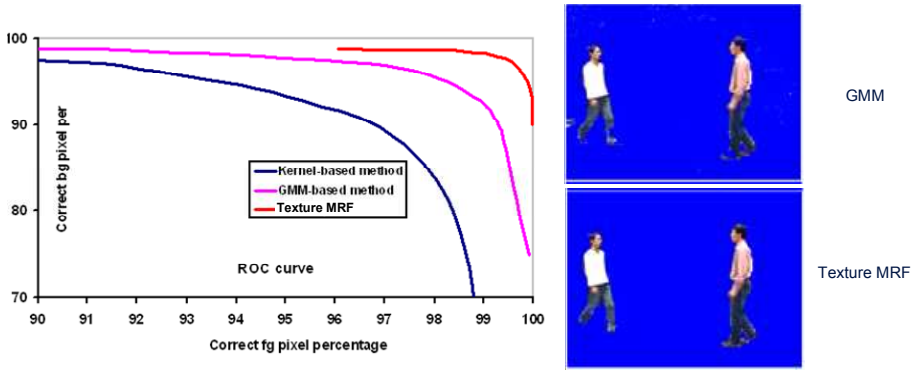

Figure 5.5: *Example of result comparison with different segmentation algorithms.*

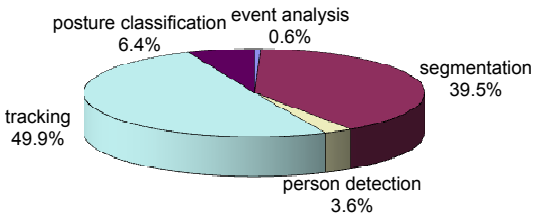

Figure 5.6: *System performance: average execution-time percentage of each module.*

5.3.3 Execution-time aspect in the system design

An increase of intelligent processing functions are integrated into an embedded system with limited hardware and software resources. These constraints limit the performance of the VCA algorithms and require design trade-offs. With the increased use of programmable hardware a set of processing functions can be implemented in software. However, the associated computations for various image processing functions are fluctuating and data dependent. Therefore, it is indispensable to address execution-time aspect for VCA functions, when the complexity of the scene and the associated processing are within the capabilities of the embedded platform.

To obtain a flexible, though powerful computing platform for VCA in a specific application, more dedicated solutions are required to improve the computation power of the platform and to be able to implement the advanced video processing algorithms. With respect to algorithm choices, the available algorithms seems to be broad and powerful enough to usefully supply the embedded applications in surveillance, etc. In addition, it requires algorithm knowledge to make efficient and powerful combinations of the algorithms for

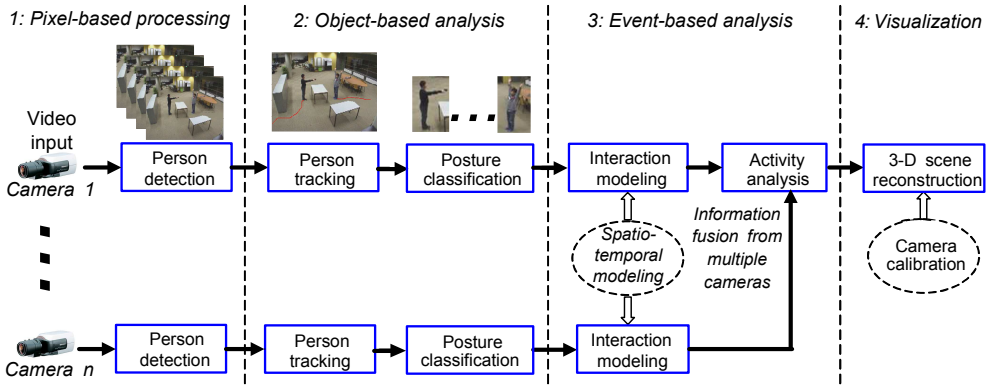

Figure 5.7: *Block diagram of our four-level motion analysis system.*

the embedded applications. Figure 5.5 shows an example of algorithm selection in the segmentation step. Although the texture MRF algorithm [25] obtains better segmentation result, it has higher complexity while it operates 2 times slower than the GMM algorithm. To guarantee the real-time performance in the surveillance application, the GMM algorithm is therefore selected while keeping the balance between the accuracy and efficiency.

To explore the execution-time difference associated with different processing blocks, an example of system performance is summarized in Figure 5.6. It presents the average cycle-consumption percentage of every module. We have assumed that camera calibration is an off-line process taking place first. As can be noticed, foreground/background segmentation (39.5%) and tracking (49.9%) consume most computing cycles. Therefore, the optimization of the resource consumption is mainly required for segmentation and tracking modules.

5.4 Proposed system framework

Our work aims at the object/scene analysis and behavior modeling of deformable objects. The proposed framework provides several layers of processing, starting from pixel-based processing, object-based processing, event-level analysis and visualization. The overall requirement is that the complete framework offers sufficiently high performance and enable surveillance applications. In the sequel, the home-care monitoring and the detection of a robbery are our key study cases.

The block diagram of our multi-level event-analysis system is shown in Figure 5.7. It consists of four different conceptual levels: *pixel-based processing, object-based analysis* (including trajectory estimation, posture classification), *event-based analysis* and *visualization*.

- **Pixel-based level.** The background modeling and object detection are implemented. Each image within the video covering an individual human body is segmented to extract the 'blobs' representing foreground objects. Afterwards, these detected blobs are refined to produce the human silhouette.
- **Object-based level.** It performs trajectory estimation and posture classification. We first track every moving person. Afterwards, a shape-based analysis is conducted to classify different posture types.
- **Event-based level.** Interaction relationships are modeled to infer a multiple-person event. This semantic analysis is thus responsible for the human activity recognition.
- **Visualization level.** With the aim of 2D-3D mapping calibration, the 3-D scene reconstruction can be conducted to visualize the scene for further analysis. This level can be simple for home use, but advanced for professional use (e.g. after crime analysis in 3-D).

The framework can be applied in single-camera or multiple-camera setups. The choice of using single or multiple cameras is basically independent on the type of surveillance applications and it is more ruled by the quality requirements or the occurrence of occlusions. If the efficiency is enforced and fast execution is highly required, a single-camera setup is conducted. To tackle the occlusion problem and improve the event-detection accuracy, multiple cameras are employed for capturing the same scene from different angles. Then the extracted information from multiple cameras is fused based on a 3-D camera calibration scheme. Finally, a 3-D scene reconstruction and communication among different cameras are achieved.

5.5 Specific issues of human motion analysis

5.5.1 Segmentation and trajectory generation

This subsection presents a brief introduction of segmentation and trajectory-generation steps involved in our motion analysis system. Figure 5.8 illustrates the block diagram of processing.

At the pixel-based processing level, the human silhouette is detected based on background subtraction. This general method can be used to segment moving objects in a scene, assuming that the camera is stationary and the lighting condition is fixed. To improve the blob segmentation, a shadow-removing approach [110] is used in our scheme. The false segmentation caused by shadows can be minimized by computing differences in a color space that is less sensitive to intensity changes.

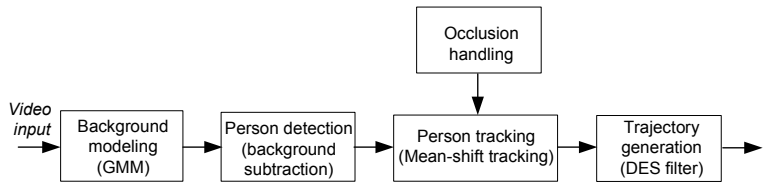

Figure 5.8: *Block diagram of segmentation and trajectory generation.*

Suppose the RGB values of a pixel with coordinates (i, j) at time t or the values of some other color-space are denoted by a vector $\vec{\mathbf{x}}^{(t)}$. In the sequel, we will leave out the coordinates for simplicity and use the short notation $\vec{\mathbf{x}}^{(t)}$. Pixel-based background subtraction involves an indication whether the pixel belongs to a background (indicated by parameter BG_{ij}) or some foreground object (indicated by parameter FG_{ij}). Again, we will leave out the pixel-based coordinates for simplicity.

Both parameters take the value zero or unity only. The estimated background model is denoted as χ_T. The estimated model is expressed by a probability measure $p(\vec{\mathbf{x}}|\chi_T, BG)$ and depends on the training set, as indicated explicitly in the conditional argument. In practice, the illumination in the scene can change gradually (variable sunlight in daytime or variable weather conditions in an outdoor scene), or suddenly (switching light in an indoor scene). Furthermore, a new object can enter the scene or a present object may disappear. In order to adapt to changes, we can update the training set $\boldsymbol{\chi}_T$ by adding new samples and discarding the old ones. We choose a reasonable time interval T starting at time t for training, so that we obtain $\boldsymbol{\chi}_T = \{\mathbf{x}^{(t)}, ..., \mathbf{x}^{(t-T)}\}$. For each new sample $\vec{\mathbf{x}}$, we update the training data set $\boldsymbol{\chi}_T$ and re-estimate the probability $p(\vec{\mathbf{x}}|\boldsymbol{\chi}_T, BG)$. However, among the samples from the recent history, there may be some values that belong to the foreground objects and we should denote this estimate as $p(\vec{\mathbf{x}}^{(t)}|\boldsymbol{\chi}_T, BG{+}FG)$. We use Gaussian Mixture Models with M components for modeling the probability measure, given by:

$$\hat{p}(\vec{\mathbf{x}}|\boldsymbol{\chi}_T, BG + FG) = \Sigma_{m=1}^{M} \hat{\boldsymbol{\pi}}_m N(\vec{x}; \widehat{\vec{\boldsymbol{\mu}}}_m, \widehat{\boldsymbol{\sigma}}_m^2 \mathbf{I}), \tag{5.1}$$

where $\widehat{\vec{\boldsymbol{\mu}}}_1, ..., \widehat{\vec{\boldsymbol{\mu}}}_M$ are the estimates of the means and $\widehat{\boldsymbol{\sigma}}_1, ..., \widehat{\boldsymbol{\sigma}}_M$ are the estimates of the variances describing the Gaussian components. The covariance matrices are assumed to be diagonal and the identity matrix $\mathbf{I}$ has proper dimensions. The mixing weights denoted by $\hat{\boldsymbol{\pi}}_m$ are non-negative and add up to unity. Given a new data sample $\vec{\mathbf{x}}^{(t)}$ at time t, the recursive update equations are calculated by

$$\hat{\boldsymbol{\pi}}_m \leftarrow \hat{\boldsymbol{\pi}}_m + \alpha(\mathbf{o}_m^{(t)} - \hat{\boldsymbol{\pi}}_m), \tag{5.2}$$

$$\widehat{\vec{\boldsymbol{\mu}}}_m \leftarrow \widehat{\vec{\boldsymbol{\mu}}}_m + \mathbf{o}_m^{(t)}(\frac{\alpha}{\hat{\boldsymbol{\pi}}_m})\vec{\boldsymbol{\delta}}_m, \tag{5.3}$$

$$\widehat{\boldsymbol{\sigma}}_m^2 \leftarrow \widehat{\boldsymbol{\sigma}}_m^2 + \mathbf{o}_m^{(t)}(\frac{\alpha}{\hat{\boldsymbol{\pi}}_m})(\vec{\boldsymbol{\delta}}_m^T\vec{\boldsymbol{\delta}}_m - \widehat{\boldsymbol{\sigma}}_m^2), \tag{5.4}$$

where $\vec{\boldsymbol{\delta}}_m = \vec{\mathbf{x}}^{(t)} - \widehat{\vec{\boldsymbol{\mu}}}_m$. Here the constant α describes an exponentially decaying envelope that is used to limit the influence of the old data. We keep the same notation, having in mind that approximately $\alpha = \frac{1}{T}$.

At the object-based level, the tracking of persons (trajectory generation) and posture classification are performed. In the trajectory-generation step, we employ the broadly accepted mean-shift algorithm for tracking persons, based on their individual appearance model represented as a color histogram. When the mean-shift tracker is applied, we extract every new person entering the scene and calculate the corresponding histogram model in the image domain. In subsequent frames for tracking that person, we shift the person object to the location whose histogram is the closest to the previous frame. After the trajectory is located, we can conduct the body-based analysis at the location of the person in every frame. When the trajectory is obtained, we can also estimate the position of the persons involved in the video scene. From our previous work [29], we have also adopted the Double Exponential Smoothing (DES) operator to track moving persons, which runs approximately 135 times faster than the popular Kalman filter-based predictive tracking algorithm, with equivalent prediction performance. After the trajectory is located, we can conduct the body-based analysis at the location of the person in every frame.

5.5.2 Individual posture recognition with CHMM

We adopt a simple but effective shape descriptor to analyze the human silhouette prior to conducting the temporal modeling scheme of a Continuous Hidden Markov Model (CHMM) to recognize the posture type.

Individual posture classification is important for human-activity recognition. First, we adopt a shape-based descriptor to analyze the human silhouette. Our posture classifier utilizes two features commonly used for object classification: area, and the aspect ratio of the bounding box attached to each detected object. This approach is simple but efficient, and it contributes significantly to the tracking and avoids a complex procedure for training data. The non-person objects and image noise can be effectively removed. The disturbance generated from different person heights is also considered. We conduct a training step regarding different heights in the scene prior to applying an adaptive threshold. Afterwards, the temporal modeling scheme of a Continuous Hidden Markov Model (CHMM) is conducted to recognize the posture

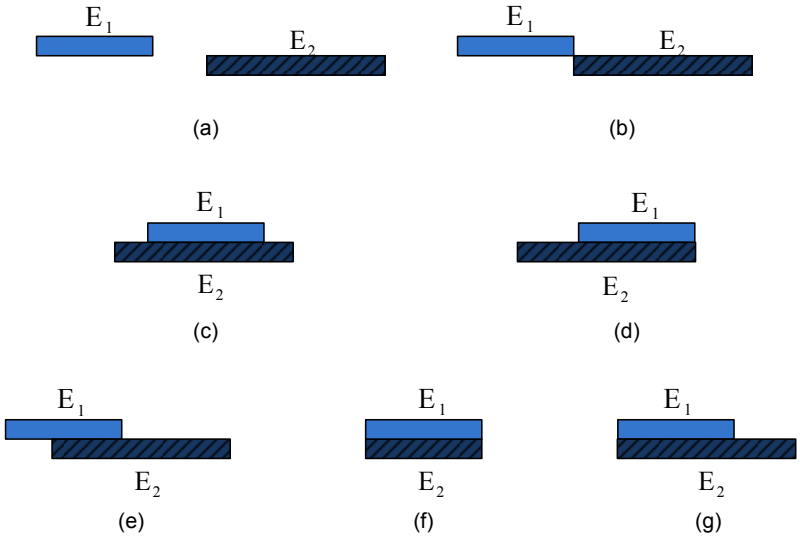

Figure 5.9: *Interaction relations between two actions E_1 and E_2 in interaction temporal logic: (a)after, (b)meets, (c)during, (d)finishes, (e)overlaps, (f)equal, (g)starts.*

type (more details can be found in Section 3.2). Finally, we can obtain the observed 2-D feature vector of the silhouette.

In our investigated case, every given posture is finally classified into one of the following types: *pointing, squatting, raising hands overhead* and *normal standing*. The above posture types are chosen because they provide important clues for scene understanding in surveillance applications.

5.5.3 Interaction modeling supporting the event classification

After the individual posture (here it is defined as an action) is classified, it is useful to further analyze the interaction between different persons, which is an important requirement for intelligent automated surveillance systems. The growing demand for safety and security has led to designing an event classifier when multiple persons are involved. In this session, we focus on the event analysis based on understanding the interactions among people involved in the scene. The temporal constraints of two-person interactions are defined by two actions in terms of causal and coinciding relations of the posture changes of the two persons. The events are seldom instantaneous and often significantly rely on the temporal order and relationship of their actions (Suppose the individual posture is the concerning action). We introduce appropriate spatial and temporal constraints for each of the various two-person interaction patterns as domain knowledge. The satisfaction of specific spatial/temporal constraints

Table 5.3: *List of temporal relations for two actions E_1 and E_2.*

Relation	Definitions and logical expressions
after	$E_2^{start} > E_1^{end}$
meets	$E_1^{end} = E_2^{start}$
during	$(E_1^{start} < E_2^{start}) \wedge (E_1^{end} > E_2^{end})$
finishes	$(E_1^{end} = E_2^{end}) \wedge (E_1^{start} < E_2^{start})$
overlaps	$(E_1^{start} < E_2^{start}) \wedge (E_1^{end} > E_2^{end}) \wedge (E_1^{end} < E_2^{end})$
equal	$(E_1^{start} = E_2^{start}) \wedge (E_1^{end} = E_2^{end})$
starts	$(E_1^{start} = E_2^{start}) \wedge (E_1^{end} \neq E_2^{end})$

contributes to the semantic recognition of the interaction. Therefore, the event-level recognition is characterized by the integration of domain-specific knowledge, whereas the object-level recognition is more closely related to the pure motion of a human body.

In order to represent temporal relationships of different actions, we apply the temporal logic based on the interval algebra of [1]. Seven temporal relationships are indicated in the set TR={*after, meets, during, finishes, overlaps, equal, starts*}. These keywords can link different actions after the individual action is analyzed. The causal and possible relations of two actions (E_1 and E_2) are illustrated in Figure 5.9. Their definitions and logical expressions are presented in Table 5.3. With this definitions and relationships, we can apply the heuristic rules to understand the scene. For example, in the application of a bank-robbery detection, the heuristic rules are based on expert knowledge. In our investigated case of robbery detection [51], the posture "pointing" is a key reference posture. It can significantly infer the robbery event. Other postures are also estimated to improve the recognition accuracy based on specific temporal constraints. An example is shown in Figure 5.10, showing the possible event of bank robbery. This event requires that person A is labeled as "pointing", person B is detected to be "raising both hands" and person C is "squatting" during the action from person A all at the same time. After performing the interaction modeling and definition of actions and events, the collecting of knowledge of action, we are able to identify the current abnormal behavior. degree of abnormality. If the value is above a predefined threshold, the surveillance system will trigger the alarm, e.g., when a detected robbery event occurs with sufficiently high possibility.

5.5.4 3-D scene reconstruction

Since we have seen in Section 2.6 that a projective transformation is equivalent to a plane-to-plane mapping, it is clear that the transform can model arbitrary

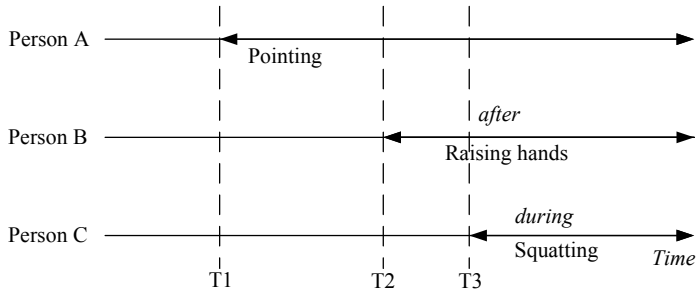

Figure 5.10: *Example of a robbery-detection case based on interactions and timing of three individual persons.*

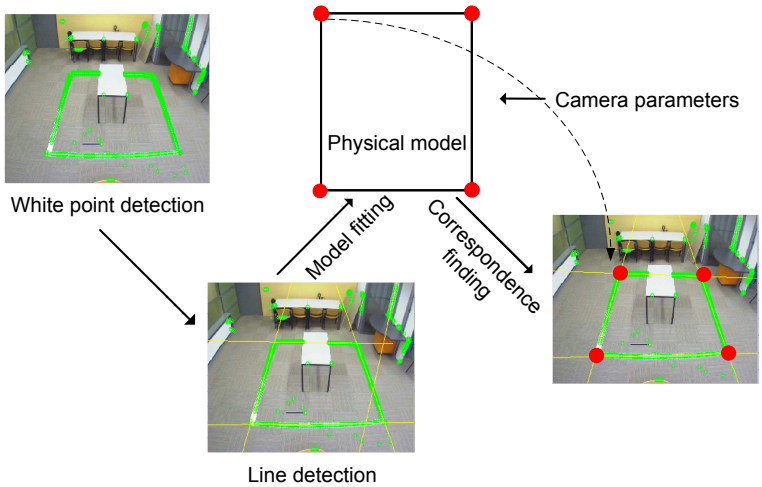

Figure 5.11: *Visualization procedure with visualization of the corresponding homography based on camera calibration.*

camera motion, as long as the observed object is planar.

The essence of the camera calibration is to provide a geometric transformation that maps the points in the image domain to the real-world coordinates. Scene reconstruction in 3-D is a useful tool in semantic-event analysis, which is generally utilized in multimedia applications [26]. An example of mapping the scene perspective using references points into a physical model is visualized in Figure 5.11. In our system, we analyze the human behavior based on the person's trajectory and/or speed on the ground, so that the height information of the human is not required. Since both the ground and the displayed image are planar, the mapping between them is a homography. The homography represents the mapping between the reference points to the 2-D image into the ground plane of the physical model (Figure 5.11). This mapping can be

specified as a 3×3 transformation matrix $\mathbf{H}$, transforming a point in real-world coordinates $\mathbf{p'} = (X, Y, Z)^T$ to the image coordinates $\mathbf{p} = (x, y, z)^T$ with $\mathbf{p} = \mathbf{Hp'}$. This transformation can be presented:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}. \tag{5.5}$$

The transformation matrix $\mathbf{H}$ can be calculated from four points whose positions are known both in the real world and in the image. In our previous work [29], we have developed an automatic algorithm to establish the homography mapping for analyzing a tennis video, where the court lines and their interaction points are identified in the image. Such lines and points are related to the lines and points in a standard tennis court. Therefore, the homography mapping described in Equation (5.5) can be established after the correspondences are found. We have conducted a similar technique in our surveillance system. The basic idea is to manually put four white lines forming a rectangular on the ground (see Figure 5.11). We have measured the length of each line in the real world, thereby defining their coordinates in the real-world domain. Afterwards, the algorithm proposed in our previous work can be applied for calculating parameters of the homography mapping. The complete algorithm comprises four steps, which are white-pixel detection, line detection, finding intersection points and calculating the parameters. For more details, we refer to an earlier publication [29].

Note that these four points need not be fixed, but should be rather selected on a case-by-case basis, as some reference points may be occluded in some views. Instead of using point features directly, we base our calibration algorithm on lines, because detecting the accurate position of a specific reference point on the ground is more difficult than estimating the position of line segments. Moreover, the detection of lines is more robust, since they are hardly occluded completely. The basic approach of the algorithm is to extract a number of straight lines from the input image, providing a set of ground candidates. Using a combinatorial search, line candidates are assigned to lines in the ground model. For each assignment, the corresponding geometric transformation can be determined. This transformation is used to project the complete ground model back to image coordinates. Each transformation is measuring the match between the back-projected model lines and the ground lines in the input image. The transformation with the best match is selected as the final solution.

After the mapping from image to real world is performed, we can estimate the position p_i and calculate the real speed of the persons v_i for the i-th person involved in the video scene. The classification into walking or standing can be therefore determined for an individual person by

$$v_i = \begin{cases} > T_{v1}, \text{person } i \text{ is running, or} \\ < T_{v2}, \text{person } i \text{ is standing, or} \\ \text{otherwise, person } i \text{ is walking,} \end{cases} \tag{5.6}$$

where T_{v1} and T_{v2} are two thresholds determining the motion types. In our experiments, we have used T_{v1}=5m/s and T_{v2}=1m/s.

5.5.5 Multiple-camera tracking of humans

The requirement for using multiple cameras for tracking arises for two reasons. First, the limitation of tracking human motion from a single view is that the field of view of a single camera can only monitor a limited area in the scene. This limitation results from several cost constraints, such as inexpensive lens and sensors. One strategy to increase the size of the monitored area is to mount multiple cameras at various locations surrounding the area of interest. The use of multiple cameras is motivated by several aspects. As long as the object is within the area of interest, it will be captured from at least one of the perspectives of the camera network. Second, tracking from multiple perspectives also helps to solve ambiguities in matching when objects are occluded from certain viewing angles. Occlusion handling is a major problem in visual human tracking. Then objects are occluded by static scenery elements, such as buildings and street lamps. However, when multiple moving objects occlude each other, especially when their speeds, directions and shapes are very similar, their motion regions merge in the image, which makes the location and tracking of objects particularly difficult. In addition, the self-occlusion of a human body is a significant and difficult problem. Therefore, the most promising practical method for addressing occlusion is to use multiple cameras.

However, compared with tracking moving humans from a single view, establishing features correspondence between images captured from multiple perspectives is more challenging. As object features are recorded from different spatial coordinates, they must be adjusted to the same spatial reference prior to matching is performed.

Most tracking techniques fail in complex situations when a lot of interacting objects are involved. Several works have studied active vision methods [91] and mainly fusion of visual information with multiple cameras [15] to improve results and robustness. This fusion is either performed on the output of different algorithms applied to the same camera [89], or on information fusion coming from the individual output of several cameras [3]. For example, Mittal [62] has addressed the problem of multi-view tracking using synchronized cameras. This system is able to combine information coming from multiple camera pairs (i.e., up to 16 synchronized cameras were used in the experiments) in order

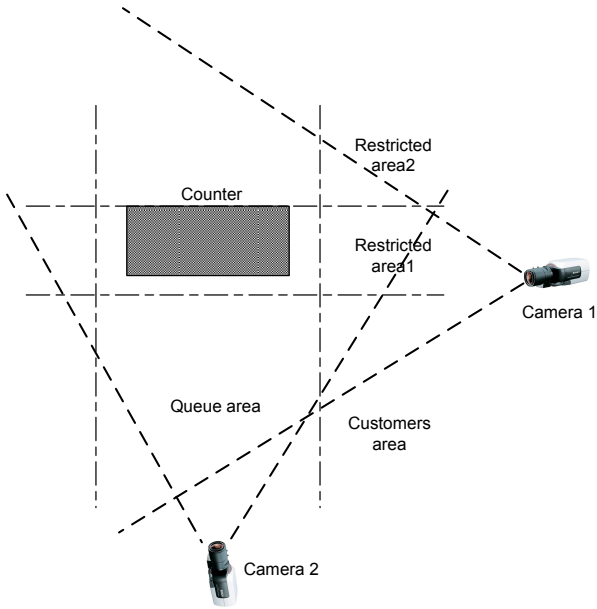

Figure 5.12: *Setup of the cameras in the experiment.*

to handle occlusions and correctly track densely located objects in a cluttered scene. However, due to the complexity, the system is not yet able to operate at real-time speed (i.e., it takes 5 seconds per processing step). The proposed system in [38] is able to track the objects in the scene using camera setup with non-overlapping field of view. First, the system learns the camera topology and the path probabilities of objects during a training phase. This is based on the assumption that people and cars tend to use the path in the surrounding. Then, the associations are performed with a maximum *a-posteriori* estimation framework.

In order to set up a real-time system for robust event detection, we intend to design a multiple-camera analyzer with the introduction of a 3-D camera calibration scheme for hierarchical and multi-view scene understanding. The camera calibration can also facilitate the information communication between different views. Its fundamental techniques refer to Section 2.6. In a multiple-camera setup, the camera calibration also provides a platform for communicating different cameras. They are connected based on a uniform real-world coordinate system. Therefore, behavior analysis from a single camera can further generate the information fusion as a whole from different viewpoints. An example of our multi-camera setup is illustrated in Figure 5.12.

Suppose the value of the person s's location with coordinates (i, j), which is captured from the w-th camera at the frame t, is denoted as P_w^t. This short notation is used after we leave out the coordinates and the person's ID

Table 5.4: *Key steps of the algorithm for multiple-camera tracking.*

Given: the number of cameras involved M and the location P_w^t for each person at the frame t.

1. Initialize the parameter $K^{t+1} = M$ at the frame $t+1$ for the w-th camera.

2. From each viewpoint, calculate the estimation of the updated location $\hat{P}_w^{t+1}$ at the frame $t+1$, by employing mean-shift tracking algorithm.

3. Conduct the occlusion detection at the frame $t+1$ for the person i, by analyzing whether the foreground objects merge in the image domain.

4. Obtain the occlusion-related parameter o_w by

$$o_w = \begin{cases} 0 & \text{if occlusion occurs,} \\ 1 & \text{if occlusion is not detected.} \end{cases} \tag{5.7}$$

5.Update K^{t+1} by
For $w = 1..M$
if $o_w = 0$, then $K^{t+1} = K^{t+1} - 1$

6. Find the updated location of the target candidate at the frame $t+1$

$$P^{t+1} = \frac{\sum_{w=1}^{M} o_w \hat{P}_w^{t+1}}{K^{t+1}}. \tag{5.8}$$

for simplicity. In summary, the key steps for multiple-camera tracking are presented in Table 5.4. Finally, the detected location for each person P^{t+1} is obtained.

5.6 Experimental results

We have conducted experiments in three different scenarios presented in the following subsections.

5.6.1 Single-camera: home-care monitoring

In our first case study on home-care monitoring, the experiment demonstrates the capability of the framework for activity classification based on the extracted location and speed of the human after performing 3-D camera calibration. Within our behavior-analysis framework, we can calculate the speed and estimate the real-world location of each individual person based on the trajectory estimation and camera calibration. The experimental videos have been

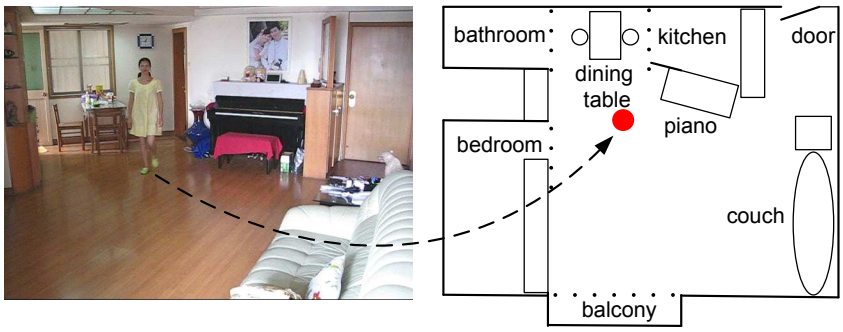

Figure 5.13: *Example image of corresponding 2-D/3-D mapping in home-care monitoring.*

Table 5.5: *The detection results for human activity recognition in home-care monitoring, the ratio m/n stands for m correct detections resulting from n experiments. (Note: A-In kitchen, B-Sitting at dining table, C-Sitting on couch, D-Playing piano, E-To balcony, F-In bedroom, G-In bathroom, H-Enter/Leave by door)*

	A	B	C	D	E	F	G	H
A	14/16	2/16	0	0	0	0	0	0
B	0	8/8	0	0	0	0	0	0
C	0	0	10/10	0	0	0	0	0
D	0	1/8	0	7/8	0	0	0	0
E	0	0	0	0	10/10	0	0	0
F	0	0	0	0	2/12	10/12	0	0
G	0	1/7	0	0	0	0	6/7	0
H	0	0	0	0	0	0	0	5/5

captured in an apartment involving 6 persons (with different gender, height, age and clothes). The length of video sequences is more than 2 hours. The layout of the apartment is illustrated in Figure 5.13. Based on the detected location in the layout after the 2D-3D mapping, the human daily activity is classified into 8 types (A-in kitchen, B-sitting at dining table, C-sitting on couch, D-playing piano, E-to balcony, F-in bedroom, G-in bathroom, and H-enter/leave by door). The classification results of the above activity recognition are summarized in Table 5.5. It is noted that the classification accuracy achieves 93.4% in the testing data sets. The classification error is mainly caused by the strong lighting reflection from the floor. This leads to segmentation errors of foreground object and its detected location is not accurate.

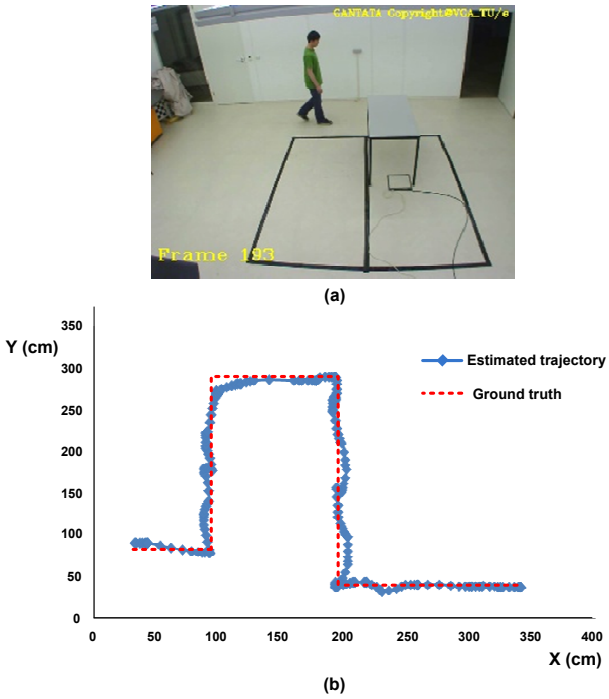

Figure 5.14: *Comparison of estimated motion trajectory of a person its ground truth in the real world.*

5.6.2 Single-camera: robbery-event detection

In our second experiment, we aim at detecting a robbery event from a monocular video. The abnormal event (a robbery detection in our investigated case) is detected based on domain knowledge (such as restricted area in the actual scene, person's abnormal moving speed, and abnormal multi-person interaction) learned at a training stage. We have trained our framework using 10 video sequences of various single/multi-person motion (15 frames/s) in a simulated robbery-event scenario. Persons with different heights, ages, clothes and individual activities (running, walking, standing which includes pointing, raising both hands, squatting and normal standing) are involved. For testing, we have applied 24 similar sequences. The system implementation features automatic camera calibration at the start of the analysis. By calculating the ratio between correct frames detected and total frames involved in the testing sequences, we have obtained a 98% accuracy rate on person detection, 95% detection rate on person tracking (where the criterion is that at least 70% of the human body is included in the detection window). In the non-occlusion situation, the posture classification (pointing, raising both hands, squatting and normal standing) is conducted. Its detection accuracy is calculated based

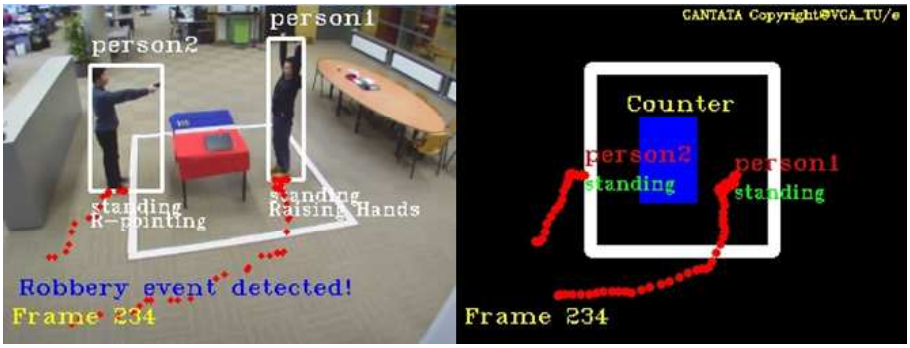

Figure 5.15: *Example of our robbery-detection result. The trajectory of every person is visualized after persons tracking. The postures are estimated and the semantic event is highlighted after interaction modeling. The camera calibration is performed and the 2D-3D mapping is visualized in the right part.*

on the ratio between the number of frames detected and the total number of the evaluated frames during testing. Employing the feature described in Section 5.5.2, the posture-classification rate is summarized in Table 5.6.

Table 5.6: *Confusion matrix for posture classification.*

Posture type	Pointing	Raising hands	Squatting	N-standing
Pointing	78%	3%	1%	18%
Raising hands	5%	86%	2%	7%
Squatting	2%	9%	83%	6%
N-standing	1%	7%	4%	88%

To evaluate the performance of the proposed 3-D reconstruction algorithm for detected person's physical location, we have captured test videos and manually measured the ground truth. Figure 5.14(a) shows a frame example selected from one of our test sequences. Figure 5.14(b) shows the 3-D reconstructed moving trajectory of the detected person and its ground truth in the physical layout of the room. It can be seen that the estimation error is very small, which is mostly less than 5%. Figure 5.15 shows an example of a simulated bank-robbery event. After the individual-posture classification and activity recognition, the semantic event is labeled on the images. The robbery detection rate is 83.3% in our captured simulated-robbery video sequences (in total 24 sequences), after performing interaction modeling.

Our run-time networked system performance was tested by video sequences

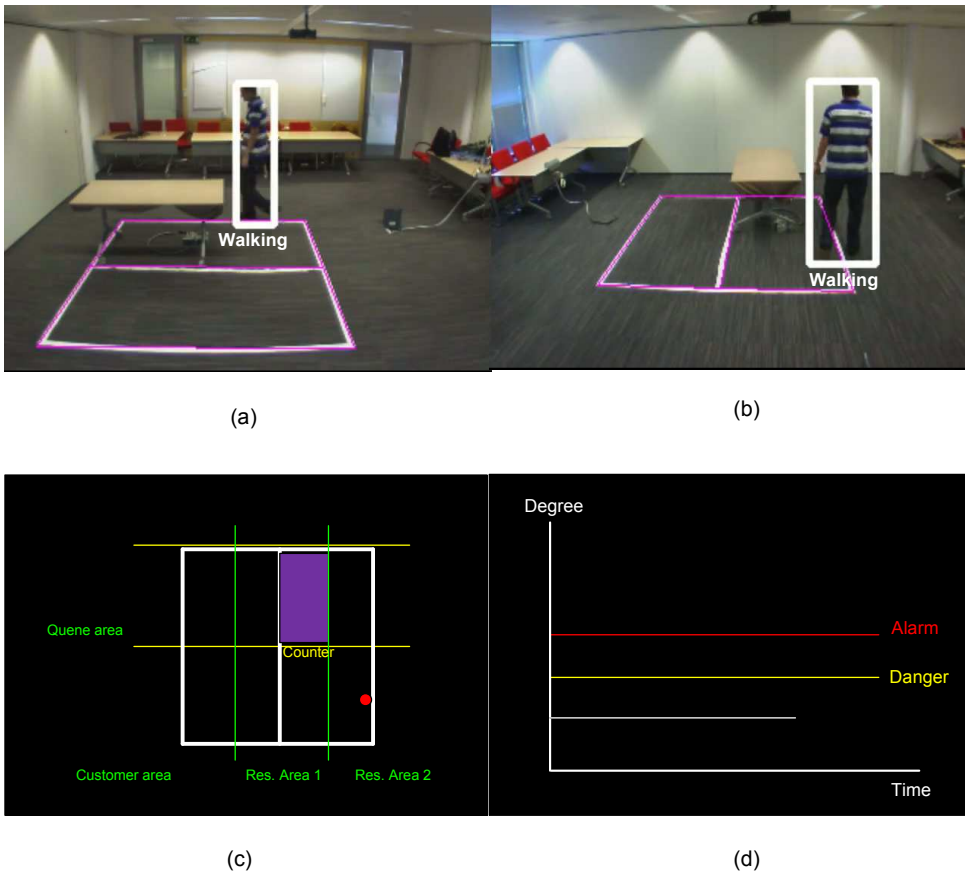

Figure 5.16: *Example of our simulated multi-camera robbery-detection result at Frame 1123.*

at 640×480 resolution (VGA), with a P-IV 3-GHz PC. We have assumed that camera calibration is an off-line process taking place first. The experiments have reviewed that a frame rate of 13-15 frames/second is obtained for monocular video sequences. This frame rate is close to real-time performance.

5.6.3 Dual-camera: robbery-event detection

To further address the problem of occlusion, multiple cameras are employed for capturing the same scene from different angles. We have conducted a dual-camera experiment within our behavior-analysis framework. Two synchronized cameras are used to capture the scene. The setup of the cameras is depicted in Figure 5.12. The testing data set is summarized in Table 5.7.

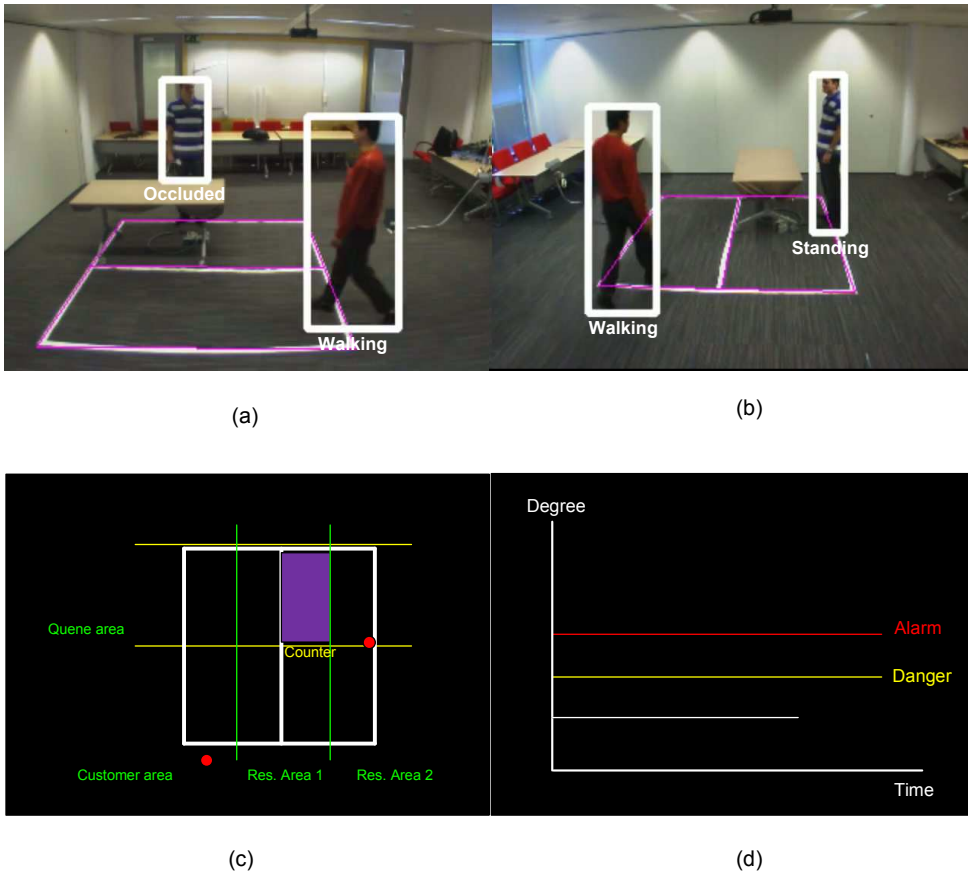

Figure 5.17: *Example of our simulated multi-camera robbery-detection result at Frame 1165.*

We analyze both camera views and combined the semantics of both views into one degree of abnormality into one measurement of abnormal behavior. Currently, a logical OR operator is applied to link the two viewpoints at the level of the abnormal-event detection. The True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN) are computed respectively. Then the event-detection accuracy is calculated by

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%. \tag{5.9}$$

The result is summarized in Table 5.8. It is noted that the dual-camera setup achieves 75.0% in accuracy rate. It is 8.3-16.7% higher than the setup with individual camera. The classification results show that the dual-camera

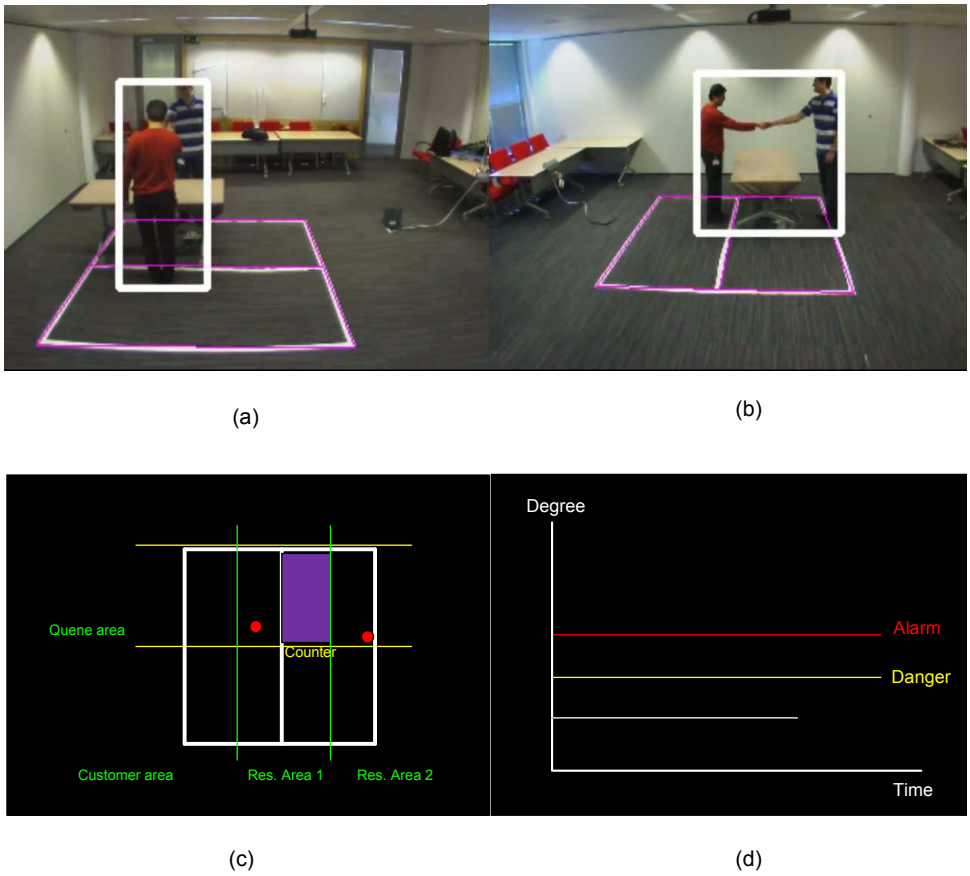

Figure 5.18: *Example of our simulated multi-camera robbery-detection result at Frame 1192.*

scheme significantly improves the event-based semantic analysis. The major reason for a possible detection error is that multi-person occlusion occurs in both viewpoints. When this occurs, the posture classification is not sufficiently accurate. Figure 5.16 through Figure 5.21 show a detection example of a simulated bank-robbery event. The position of every person is visualized after trajectory generation. The postures are estimated and the semantic event is highlighted after interaction modeling from two different viewpoints. The degree of abnormal behavior is calculated and graphically visualized in each figure. The camera calibration is performed and the resulting 2D-3D mapping is visualized. Although the posture pointing is not recognized in one camera (see Figure 5.21(a), it is correctly recognized in the other camera (see Figure 5.21(b)), so that the robbery event is successfully detected. This system

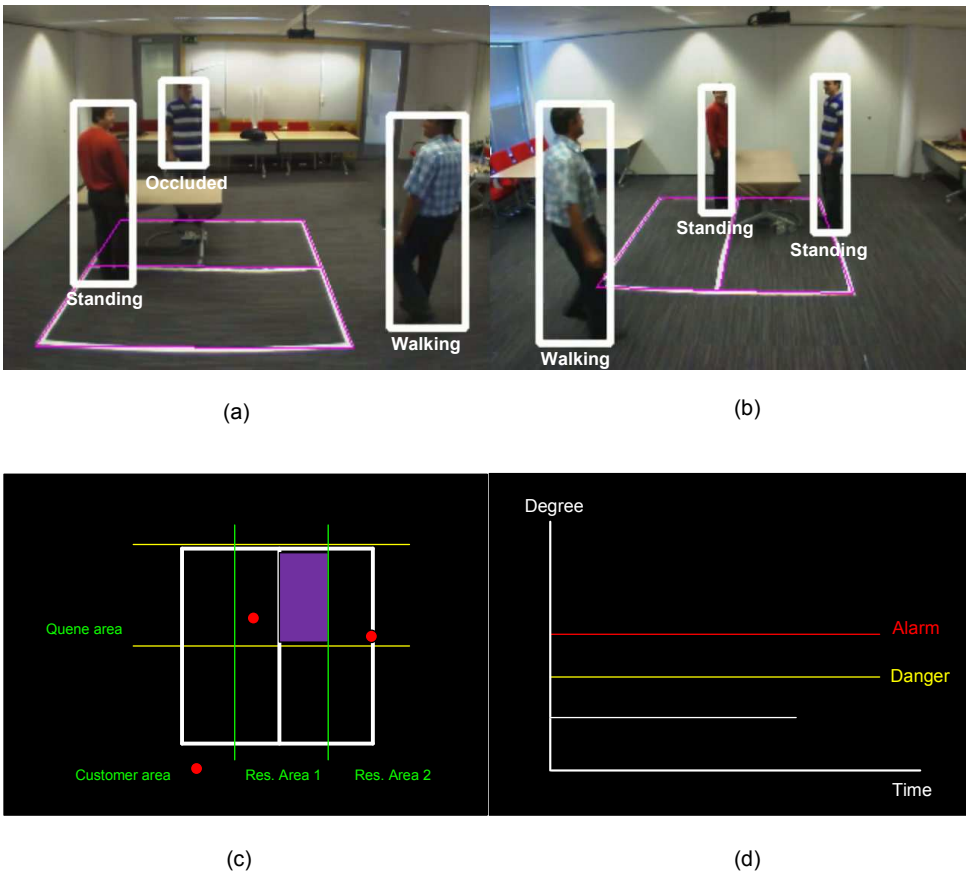

Figure 5.19: *Example of our simulated multi-camera robbery-detection result at Frame 1279.*

was inserted in a component-based architecture and successfully executed for live demonstrations in the European ITEA project CANTATA. The whole system was executed on a P-IV PC running at 3 GHz. In our experiments, we have achieved a 6-8 frames/second frame rate for two video sequences.

5.6.4 Extension to street-fighting game application

Our proposed multi-level framework can be extended to other applications, e.g. interactive gaming in augmented reality. The techniques for the surveillance application are reused in a new video-based scenario of an interactive street-fighting game. A visual example of the result is shown in Figure 5.22. The new virtual environment image is defined by the user. The original frame

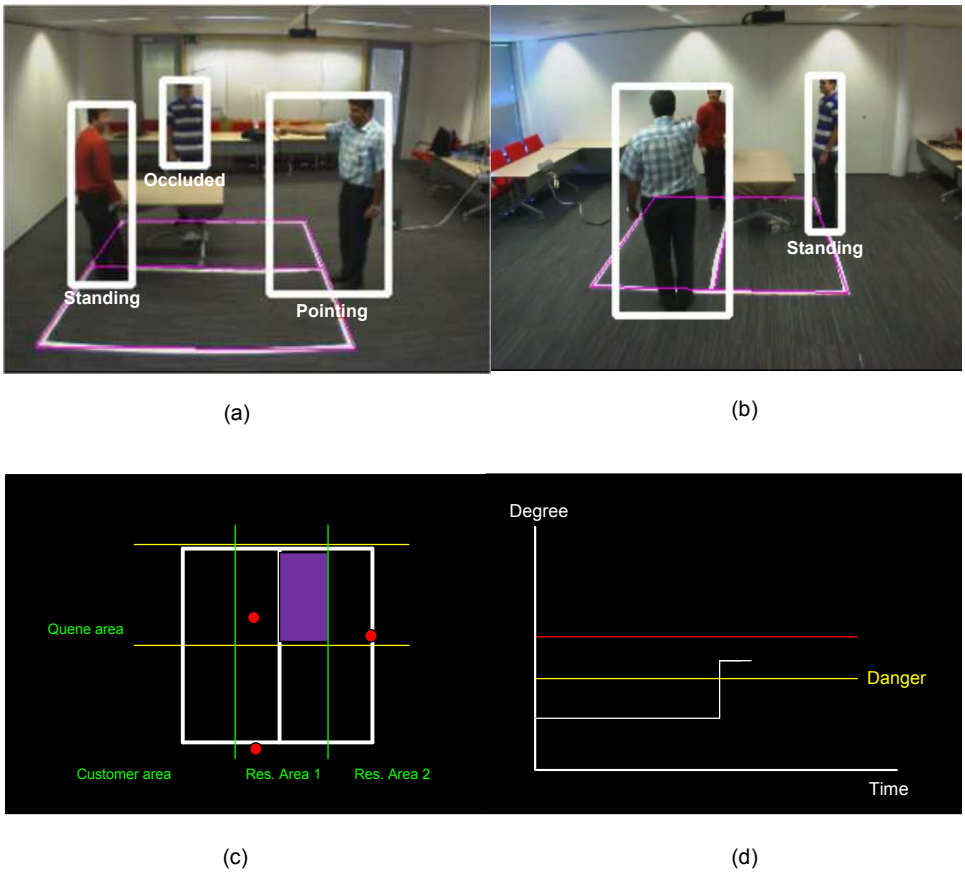

Figure 5.20: *Example of our simulated multi-camera robbery-detection result at Frame 1304.*

involving players is online captured by the camera and shown in Figure 5.22(a). The result of the augmented reality scene of reconstruction is shown in Figure 5.22(b). Based on the detected position of players' hands, virtual attacking blocks in the form of red squares are moving at a predefined speed in the horizontal direction of the detected hands. Then the players' score is calculated based on the times when the flying block hits the opponent in the image. Finally, the score bar for each person is visualized in the output frame.

5.7 Conclusions

This chapter has discussed various aspects for the design of a complete framework aiming at efficient event understanding in surveillance applications. First,

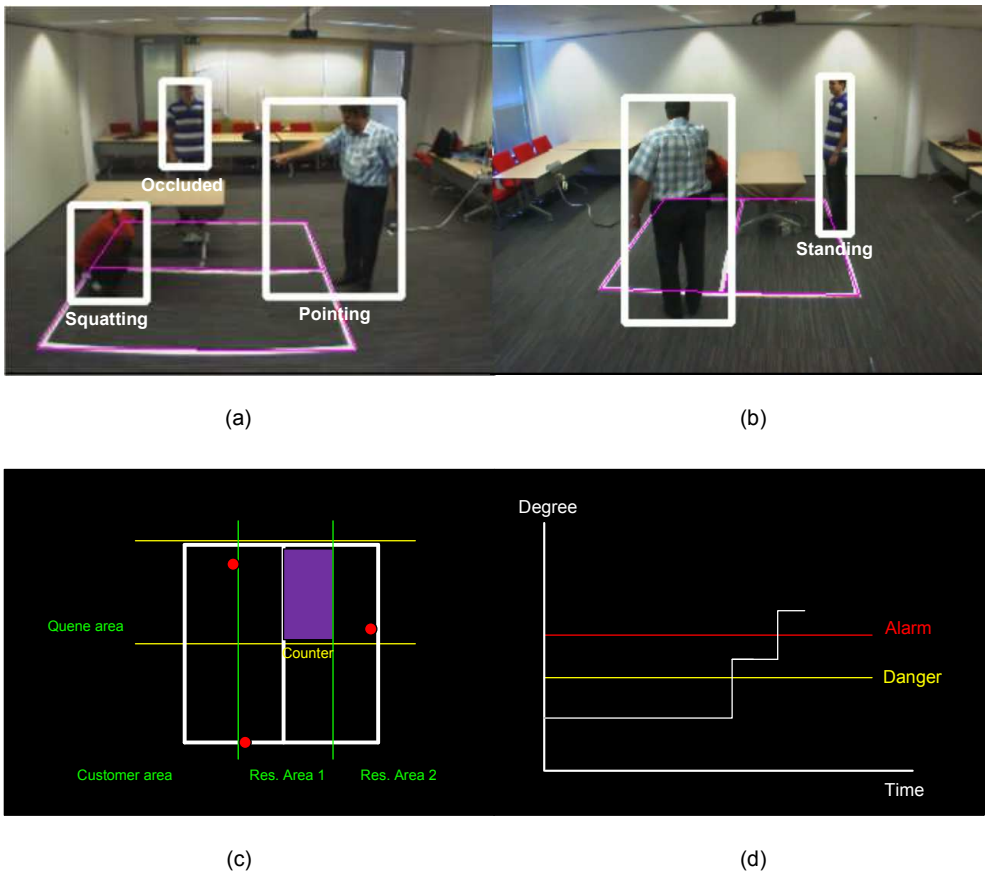

Figure 5.21: *Example of our simulated multi-camera robbery-detection result at Frame 1323.*

it reviews existing surveillance systems and summarizes requirements of surveillance analysis systems. Then, we have introduced our system design based on a specific software architecture, by employing the well-known "4+1" view model [77]. This model describes a software architecture using five complementary views, each of which addresses specific concerns. For example, in the physical view, the installation of a camera network and the setup of the calibration grid in our surveillance application is carefully considered.

To design an efficient system, human behavior analysis algorithms are directly designed for real-time operation and embedded in an experimental run-time AV content-analysis architecture. This run-time architecture is designed to be generic for multiple streaming applications with a *component-based* architecture design. It facilitates advanced video content analysis for surveillance,

Table 5.7: *The testing data set involved in event detection based on single/multiple-camera setups. (Note: A-Running, B-Walking, C-Pointing, D-Raising hands overhead, E-Squatting)*

ID	Scenario	No. of persons	Ground truth	Cam1 detection	Cam2 detection	Cam1+Cam2 detection
1	A	1	Yes	Yes	Yes	Yes
2	A	1	Yes	Yes	Yes	Yes
3	A	2	Yes	Yes	Yes	Yes
4	B	1	Yes	Yes	No	Yes
5	C+D	2	Yes	No	No	No
6	C+D	2	Yes	Yes	Yes	Yes
7	C+D+E	3	No	Yes	No	Yes
8	C+D	2	No	No	No	No
9	C+D	2	Yes	Yes	No	Yes
10	C+D+E	3	Yes	No	Yes	Yes
11	C+D	2	No	Yes	Yes	Yes
12	C+D	2	Yes	Yes	No	Yes

Table 5.8: *The detection results for abnormal event.*

	TP	TN	FP	FN	Accuracy
Cam1	7	1	2	2	66.7%
Cam2	5	2	1	4	58.3%
Cam1+Cam2	8	1	2	1	75.0%

featuring network-based communication. We utilize the modified ViPER file format to represent XML data within the architecture. First, the frame-span attribute of each object is replaced by a time-stamp attribute. Second, the Video Content Analysis (VCA) component sends new or updated data on objects, e.g. location and action, without data redundancy. Furthermore, the execution-time aspect is considered in the system design. Although the texture MRF algorithm [25] obtains better segmentation result, it has higher complexity, as it operates two times slower than the GMM algorithm. To guarantee the real-time performance in the surveillance application, the GMM algorithm is selected and executed in the CANTATA online demo in Madrid. Besides this, the average cycle-consumption percentage of each processing module is calculated. We have found that foreground/background segmentation (39.5%)

(a) (b)

Figure 5.22: *Extensibility to application of interactive street-fighting game: (a) the original frame captured from the camera; (b) the detected person with fists and score superimposed in the new user-defined background image.*

and tracking (49.9%) modules consume most computing cycles.

Afterwards, we have proposed a layered framework that enables multi-level human motion analysis, featuring layers at pixel, object, event and visualization level. At the object-based level, both trajectory generation and posture classification are performed. We first track every moving person. Later, a shape-based analysis is conducted to classify different posture types. At the event-based level, interaction relationships are modeled, based on temporal logic between two different actions, to infer a multiple-person event. This semantic analysis is responsible for the human activity recognition. At the visualization level, the 3-D scene reconstruction is conducted to visualize the scene based on 2D-3D mapping technique. This level can be simple for home use, but advanced for professional use (e.g. after crime analysis in 3-D).

Our proposed layered framework has been implemented in software and embedded in a run-time architecture and we have first applied this system architecture with a single camera. The experimental platform operates at two Pentium Quadcore (2.33 GHz) and 4 GB memory. Performance evaluations have shown that this networked framework is efficient and achieves a fast performance, of about 13-15 frames/second, for a monocular video sequence.

In addition, we have tested a dual-camera setup within the behavior-analysis framework. Based on our automatic camera calibration scheme, the 3-D reconstruction and communication among different cameras are achieved. It is possible to benefit from the extra camera view in case of occlusion and it may also add to after-crime analysis. We have proposed a multi-camera tracking algorithm. If occlusion is detected for an individual camera view, the tracking is conducted based on another view. If occlusion is detected for both

views, the moving person's location is estimated from preview frames. The extension of multiple-view fusion improves the event-based semantic analysis by 8.3-16.7% in accuracy rate. Moreover, we achieve a 6-8 frames/second frame rate for two video sequences with a P-IV PC running at 3 GHz. With optimization and multi-streaming implementation, our designed multi-camera setup for behavior analysis can be further used in a real-time system implementation.

Furthermore, our proposed behavior analysis system can be extended to interactive gaming application, which shows the extensibility of our applied motion analysis techniques.

This chapter has discussed how to integrate specific domain knowledge and we have employed 3-D camera calibration to support further semantic analysis of the scene in surveillance applications.

Chapter 6

Conclusions and Outlook

6.1 Conclusions of each chapter

In this thesis, we have addressed selected improvements in human behavior analysis and two complete real-time applications based on such behavior analysis. Essentially, the thesis consists of three parts. Chapter 2 and Chapter 3 describe some fundamental techniques and provide an overview of visual human motion analysis and individual person analysis, respectively. Chapter 4 and 5 employ these analysis techniques and describe the created full applications for tennis sports and surveillance. These chapters contain our major new contributions. In the following, we summarize the contents and findings of each individual chapter.

In **Chapter 1**, we have introduced the background of human motion analysis and have motivated the research contributions of this thesis. We introduce a generic motion-analysis framework consisting of three processing-stages: pixel-based processing (background modeling and segmentation), object-based modeling (human tracking and posture analysis), and event-based analysis (semantic event understanding). We have discussed the requirements such as efficiency and extensibility for designing smart systems using human behavior analysis, e.g. in the consumer and security domain. After presenting the problem statement, we summarize our contributions and provide the scientific published papers that form the background of the succeeding chapters.

Chapter 2 presents an overview of state-of-the-art motion-analysis techniques, which cover the various, different processing techniques required to perform human motion analysis. These techniques are background modeling,

human-body profile detection, human tracking and behavior understanding. We discuss a generic diagram addressing three levels of techniques within a general processing framework: pixel-level processing, object-level modeling and event-level analysis. We have found that Gaussian Mixture Models (GMMs) are suitable for fast algorithm deployment, while offering sufficient accuracy. These models can be used to segment the players at the foreground within the tennis court. We have used features like *shape* parameters and *silhouette* to distinguish a human from other moving objects. We present several techniques for human tracking and focus on region-based tracking. At the event-based modeling, the constraint-based approach is simple to implement and suited for real-time operation. For the behavior analysis, we have taken a mathematical solution in the form of analyzing the timeline of individual person of interaction of those persons.

In **Chapter 3**, we have discussed the techniques involved in motion analysis of each individual person. This chapter presents a multi-module framework for the multi-level analysis of human motion. The framework captures the human motion, classifies various postures and perform body-part detection.

First, we discuss a layered framework for posture modeling and representation. We contribute with an efficient HV-PCA shape-based descriptor using horizontal and vertical projections, and obtain more accurate and compact representation. The proposed HV-PCA descriptor combined with temporal modeling achieves a posture-recognition accuracy rate of about 86% and outperforms other existing proposals. Our human motion scheme is optimized for efficient operation offering a fast performance (6-8 frames/s), to enable further analysis of human behavior in a surveillance application.

Second, we have proposed a novel approach for body-part detection. The trajectory-based estimation and body-based analysis are combined simultaneously to capture the human motion and locate the different body parts. The trajectory-based module performs the human tracking. The body-based module infers the posture of the human body and presents the body geometry efficiently by a skeleton model. To locate reliably the skeleton model, both Nearest-Neighbor filtering and a tracking filter are employed. In addition, we have presented a new algorithm for accurately locating the body center point, based on the body silhouette and an upper/lower-body separation line. This algorithm outperforms the conventional center-of-gravity approach from existing literature, addressing the same center-point usage. Furthermore, the conventional assumption of upright body posture is not required. The proposed scheme achieves a near real-time speed (10 frames/s) within monocular video sequences in indoor/outdoor areas.

Chapter 4 develops a tennis sports video analysis system, featuring high-level scene analysis based on real-world visual cues. First, playing-frame detection involves the selection of the tennis playing field sequences out of a full broadcast sports program, including special scenes, like interrupting commer-

cial breaks. Afterwards, court detection identifies the court location in the scenes and provides its specific information, such as size and shape. Then, segmentation of players and a subsequent tracking module calculate the position and speed of each player, which are required to derive individual player behavior and tactic. Afterwards, the camera calibration deduces a semantic meaning from the position and movements of the players, which takes the camera motion into account and computes the player positions in the real-world coordinates, based on the 2D-3D transformation. Finally, we perform semantic analysis, classify different events (e.g. service, base-line rally and net-approach) and derive a game abstract showing various types of information, such as real running distance and behavior of the players. In this chapter, a main contribution is that several selected real-world visual cues are applied to a set of linear-weighted models of individual events. Timeline functions are additionally exploited to achieve robust event detection, as probabilistic values are produced to indicate the reliability of the event occurrence. Experiments show that the complete proposed application is efficient enough to obtain a real-time or near real-time performance (2-3 frames/second for 720×576 resolution (4CIF), and 5-7 frames/second for 320×240 (CIF) resolution, with a P-IV PC running at 3 GHz). Furthermore, the system may be extended to analyze other sport types [32], like badminton, volleyball and basketball, where many functions can be reused, like court detection, camera calibration, player segmentation and high-level analysis.

Second, we have presented a scheme for automatic audio analysis of a broadcasted sports program which employs specific game context knowledge. We contribute with a three-step racket-hit detection algorithm to accurately classify various events. After the refinement step, this algorithm applies a linear combination of short-time energy and spectral power. As the third step, heuristic rules, based on specific knowledge of the tennis game, are employed for mapping between the sample-level features and the semantic level. The system implementation can classify various events such as rally, scoring, different types of service, and return. The experimental results show that a detection accuracy up to around 90% is achieved for racket-hit detection.

Finally, we have presented a scheme for automatic tennis-ball-path inference for tennis sports video analysis, based on simultaneously combining audio and video clues. For example, the audio analysis reveals the service point which is combined with detecting the service player and its location in the video. The main contribution is that the tennis-ball-path is generated for tennis game tactics analysis without detecting and tracking the ball itself, by projecting players in a real-world model and combining previous analysis with game rules. The experiments showed that our system can deliver a tactical analysis based on the fusion of video and audio analysis modules.

In **Chapter 5**, we have discussed various aspects for the design of a complete framework aiming at efficient event understanding in surveillance appli-

cations. First, we have introduced our system design based on a specific software architecture, which is reported in the well-known "4+1" view model [77]. Different complementary views are analyzed, like physical and network views. In the system, human behavior analysis algorithms are directly designed for real-time operation and embedded in an experimental run-time AV content-analysis architecture. This run-time architecture is designed to be generic for multiple streaming applications with a *component-based* architecture design. We utilize the modified ViPER file format to represent XML data within the architecture, where we have carefully considered execution time in the system design. Compared with the-state-of-the-art segmentation algorithm texture MRF [25], the GMM algorithm operates two times faster, so that the GMM algorithm is chosen in the surveillance application, although texture MRF provides relatively better segmentation results. Moreover, the average cycle-consumption percentage of each processing module is calculated. It is noted that foreground/background segmentation and tracking modules are the most expensive and consume 39.5% and 49.9% of computing cycles, respectively.

A layered, flexible human motion analysis framework captures the human motion, classifies its posture, infers the semantic event exploiting interaction modeling, and performs the 3-D scene reconstruction. To evaluate the performance, the framework is embedded in a run-time architecture and we have applied this networked system in a single-camera setup. The experimental platform operates at two P-Quadcore CPUs (2.33 GHz) and 4 GB memory. Performance evaluations have shown that this networked framework is efficient and achieves a fast performance (13-15 frames/s) for a monocular video sequence. Moreover, a dual-camera setup is tested within the behavior-analysis framework. After automatic camera calibration, the 3-D reconstruction and communication among different cameras are achieved. The extra view in the multi-camera setup improves the human tracking and event detection in case of occlusion. This extension of multiple-view fusion improves the event-based semantic analysis by 8.3-16.7% in accuracy rate. As we achieve a 6-8 frames/s efficiency for two-view video sequences with a P-IV PC running at 3 GHz, we conclude that the multi-camera system and behavior analysis can be implemented as an embedded system after some further optimizations.

6.2 Discussion on research questions

Let us review the three research questions that we have posed in Chapter 1 of this thesis. The following paragraphs describe how we have addressed these research questions in this chapter.

RQ1:*How can we efficiently represent the human body in order to facilitate real-time behavior analysis?*

In Chapter 3, we have addressed the problem of individual behavior analysis of a single human, involving action recognition and body-part detection. The contributions in this chapter are both on detecting body parts and efficient body representation. We have proposed a novel body representation based on a silhouette feature. This feature is obtained by analyzing the horizontal and vertical projection of a human body's silhouette. A main advantage of this representation is that only pure binary-shape information is used for posture classification without texture/color or any explicit body models, so that it is fairly robust with limited complexity. After temporal modeling, we achieve an accuracy rate of about 86% for posture recognition, which outperforms other existing proposals. The developed human motion scheme is efficient achieving a fast performance (around 10 frames/s), and is suited for broader surveillance applications. In addition, to find the minimum amount of features for detecting human body, we present a scheme with only three features (body ratio, shape, color). This simple scheme is generic and integrated into a fast framework for body-part detection, without the conventional assumption of the human's posture being upright.

RQ2:*How should we efficiently use 3-D modeling for improved scene understanding?*

The use of multiple cameras is very helpful for a robust understanding of the human behavior, because the human body can be observed from multiple directions and occluding situations in one camera can be circumvented by using another camera view. It is beyond doubt that this will improve the behavior interpretation and event classification. The use of multiple cameras allows to reconstruct the scene in 3-D space, so that the position of objects can be calculated and a different view of actual events can be presented. More specifically, an efficient 2D-3D mapping has been solved by incorporating a fast 3-D camera calibration algorithm which was adopted from joint earlier work in the research group. This 3-D mapping then enables a computation of location and speed of objects and more detailed scene visualization, where these data can be used to create a top view of the scene where the motion and position of objects is shown for analysis purposes. This feature can significantly contribute to the scene understanding, like after-crime analysis and health-care behavior analysis of people. In this thesis, we have provided a scheme with integrated 2D-3D mapping for both tennis sports analysis and surveillance applications.

RQ3:*How can we implement behavior analysis for a complete application and facilitate real-time execution?*

For consumer and embedded applications, a (near) real-time performance is generally required while using low-cost hardware. Therefore, the algorithms have to be executed with limited processing resources and capacity. It means that algorithms have to be efficient and their complexity is constrained. Real-time execution is indispensable for surveillance applications, where the behavior sometimes should lead to direct action. In sports applications, the system should be fast enough to understand the proceeding of the game.

In Chapter 4, we have proposed a near real-time AV-based tennis sports analysis system, featuring high-level scene analysis based on real-world visual or audio cues. The real-time execution has been considered by using relatively simple cues and computations and considering data exchange with low-complexity interfaces. For example, playing field detection is based on white line detection and filtering, instead of applying advanced transforms. Computationally rich functions like a calibration algorithm was optimized for fast performance. Moreover, the automatic sports analysis system can generate metadata and with modelling, this supports faster categorizing of sports videos and fast searching and retrieval of specific sports video. In this chapter, audio signals are also used but with moderate complexity functions, this leads to higher event detection scores and robustness.

In Chapter 5, we have constructed an experimental real-time video-analysis system based on analyzing events with one or two cameras. This system was inserted in a component-based architecture and successfully executed for live demonstrations in the European ITEA project CANTATA. Here the efficient communication in a specific dedicated file format within the architecture, together with the simple modeling and associated data exchange between functions has ensured highly efficient execution of the complete application. Also in this application, human behavior analysis algorithms are specifically designed for real-time operation, such as the GMM algorithm chosen in the segmentation. The whole system has been executed on a single regular PC, so that it is feasible for mapping into an embedded application.

6.3 Future work

The holy grail of visual motion analysis is a system that can accurately interpret the motion of a human wearing clothing of any description and under varying lighting conditions with a camera that is moving and tracking the subject. The system should work in real time and provide feedback regarding the visual accuracy to its users. This system can even recognize parts of the motion as identifiable known gestures, and be able to label or interpret accordingly. At the time of writing this thesis, this goal is still to be achieved.

Despite the successful applications of our work as described in the previous chapters, we identify several interesting aspects that need further investigation.

1. Fusion of data from multiple sensors
It is obvious that future visual surveillance systems will greatly benefit from the use of multiple cameras. It is interesting to utilize other sensors including audio, infrared, ultrasonic, and radar, etc. Each of these sensors has its own characteristics. Surveillance using multiple different sensors seems to be a very challenging subject. The main problem is how to make use of their respective merits and fuse information from such kinds of sensors.

2. New hardware and software embedding implementations
In this thesis, computation efficiency is an important performance objective in the algorithm design. It is interesting to further exploit implementation issues to deploy the algorithms on suitable embedded platforms. For example, for mobile applications, it is interesting to exploit very efficient mapping of algorithms on DSP media processors. It is useful to optimize algorithms for embedded processing in cameras to facilitate intelligent video surveillance, e.g. home-use entrance control. The developed system should be extensible enough, be based on standard hardware and exploit plug-and-play technology.

3. Real-time scene reconstruction in 3-D space
The growing demand for safety and security has led to more research in building more efficient and intelligent automated surveillance systems. Therefore, a future challenge is to develop a 3-D real-time scene reconstruction, which is able to perform with minimal manual reconfiguration for various applications. Such systems should be sufficiently adaptive to adjust automatically and cope with changes in the environment like lighting, scene geometry or scene activity. An example of such a modeled presentation is portrayed by Figure 6.1. The realistic visualization shows some extensibility of our proposed framework in this thesis, which significantly facilitates the user understanding of the actual scene. However, with the trend in surveillance towards HD resolution imaging, a system design for full 3-D understanding would certainly need a notable number of extensions and efficiency optimizations would again be necessary to safeguard acceptable system costs.

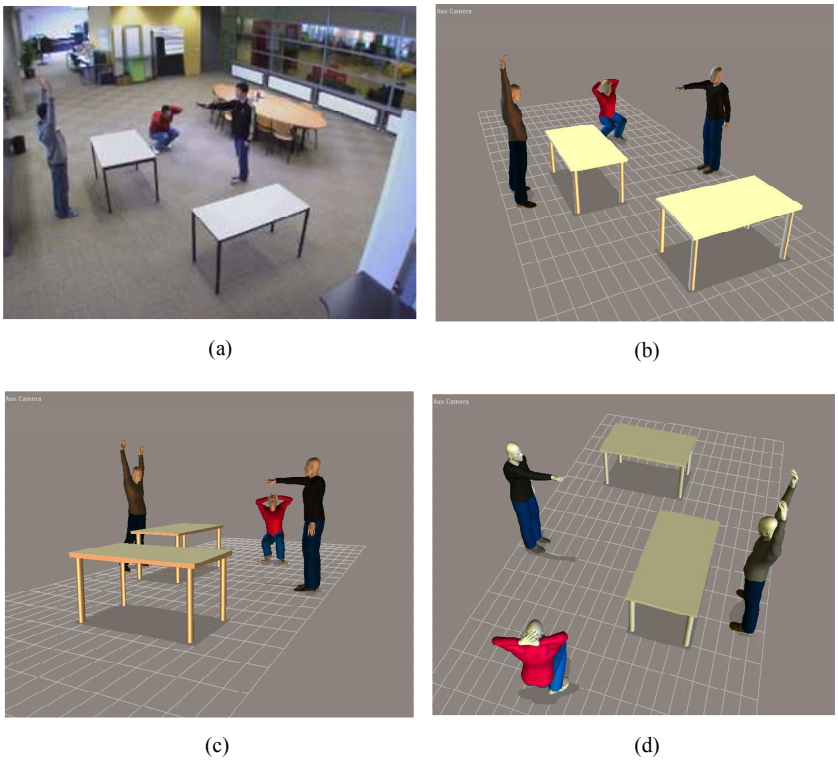

Figure 6.1: *Example of the reconstruction result of multi-person activity in a simulated bank-robbery scene: (a) original frame, (b)-(d) three different viewpoints and focal lengths to show the reconstructed 3-D scene.*

Appendix A

Appendix

A.1 Projective geometry

The geometric relations between objects in the 3-D world and a 2-D image is of central importance when we want to estimate the motion of the camera from a sequence of images. In particular, we need a geometric model that describes the observed motion fields captured by cameras. Object motion can be presented mathematically with the concept of projective geometry.

The theory of projective geometry establishes the basis for 3-D computer vision and computer graphics. Compared to Euclidean geometry, projective geometry facilitates the description of rigid 3-D motion and the perspective projection onto planar images, because it enables to formulate both with linear algebra techniques. In particular, projective geometry provides a uniform description of situations that require special cases in Euclidean geometry, like the intersection of parallel lines. Consequently, projective geometry has become the standard technique to describe 3-D geometry.

Projective geometry serves as a mathematical framework for 3-D multiview imaging and 3-D computer graphics. It is used to model the image formation process, generate synthetic images, and reconstruct 3-D objects from multiple images. To model points, lines or planes in a 3-D space, the Euclidean geometry is usually employed. However, a disadvantage of the Euclidean geometry is that points at infinity cannot be modeled and are considered as a special case. This case can be illustrated by using a perspective drawing of two parallel lines. In perspective, two parallel lines meet at infinity at the vanishing point. However, the intersection of the parallel lines at infinity is not easily modeled by the Euclidean geometry. A second disadvantage of Euclidean geometry is

that projecting a 3-D point onto an image plane requires a perspective scaling operation. As the scale factor is a parameter, the perspective scaling requires a division that becomes a non-linear operation. Therefore, the use of the Euclidean geometry should be avoided.

Projective geometry constitutes an attractive framework to circumvent the above disadvantages of the Euclidean geometry. In Euclidean space, a point defined in three dimensions is represented by a 3-element vector (X, Y, Z). In the projective space, the same point is described using a 4-element vector $(X_1, X_2, X_3, X_4)^T$ such that

$$X = \frac{X_1}{X_4}, Y = \frac{X_2}{X_4}, Z = \frac{X_3}{X_4}, \tag{A.1}$$

where $X_4 \neq 0$. In general, the coordinates $(X, Y, Z)^T$ and $(X_1, X_2, X_3, X_4)^T$ are called *inhomogeneous* coordinates and *homogeneous* coordinates, respectively.

As a generalization, the mapping from a point in the n-dimensional Euclidean space to an $(n+1)$-dimensional projective space can be denoted as

$$\underbrace{(X_1, X_2, ..., X_n)^T}_{\textit{Euclidean space}} \rightarrow \underbrace{(\lambda X_1, \lambda X_2, ..., \lambda X_n, \lambda)^T}_{\textit{Projective space}}, \tag{A.2}$$

where $\lambda \neq 0$ corresponds to a scaling parameter λ. This scaling parameter is often called the homogeneous scaling factor. Using the presented projective geometry framework.

While this appendix only gives a brief introduction, a thorough discussion of 3-D geometry, estimation of camera parameters, and multi-view geometry can be found in the book [34].

A.2 ViPER file format

The ViPER format is an initiative of the Language and Media Processing Laboratory, University of Maryland. ViPER is an abbreviated name of The Video Performance Evaluation Resource. It is originally the system for evaluating video content analysis [58]. ViPER has been originally designed to enable the evaluation of video analysis like tracking people, detecting text, etc. ViPER is not only a toolkit of scripts and JAVA programs that enable the markup of visual data ground truth, but also a system for evaluating how close between sets of result data and their corresponding ground truth.

The ViPER toolkit is composed of:

- A graphical authoring tool, ViPER-GT. This program allows frame-by-frame markup of video metadata stored in the ViPER format. It can also be used for visualization of results of video content analysis (VCA);

- A command line tool for evaluation of VCA performace, ViPER-PE;
- Other tools like the ViPER API which provides a set of Java interfaces and classes to access data stored in the ViPER format.

Let us now describe the structure of the ViPER file format. The ViPER file format is an XML format, the root element of its XML structure is <viper>. There are two sections in this element, <config> and <data>. The *configuration* section defines the descriptors and the *data* section instantiates descriptors for recognized objects or persons. The configuration section contains the definition of descriptors and it also permits using ViPER tools. A descriptor in this section is defined as follow:

- It is a record that describes some elements of a video sequence.
- It is an object that conforms to a user defined schema.
- It is composed of several defined attributes.
- It has a unique ID and an associated span which is valid.
- It is one of three types: *File*, *Content*, or *Object*:
  File:
  It refers to data that reflects the video as a whole or other metadata about the video, such as file format and frame rate.
  Content:
  Instances of this type may only occur once at a time, and any given instance may not change over the course of its life. Each instance has a time span and a set of attributes.
  Object:
  It refers to an object that may have many instances at any given time, and the instances may change over time.

The data section instantiates descriptors and contains objects and people description. For each object or person identified in the video, an <object> element is used with the following attributes:

- framespan: string, named framespan to comply with ViPER specification but could be just a frame number, e.g. framespan="12:14" or framespan="12";
- ID: a non-negative integer, which is the identifier of a specific object. It should be conserved to identify the same object in all frames;
- name: OBJECT or PERSON.

The element <object> should contain an element which indicates the object's spatial location. There are several attributes in the ViPER file format:

bbox

The bbox is a non-oriented rectangle shape. It can only be positioned with the sides parallel to the axis of canvas. The string representation of a bbox, used in the spreadsheet view, is x, y, h, w, where x and y are the coordinates of the top-left corner of the box, h is the height and w is the width. It should be noted that the coordinates of the image count down and to the right from the top left corner.

obox

The obox is the oriented rectangle. It offers the features of the bbox, with the addition of a fifth number, the orientation. The string representation of an obox is $(x, y), h, w, o$, where x and y are coordinates of the top-left corner of the box, h is the height, w is the width and o is the orientation (angle) in counter-clockwise degrees.

ellipse

The ellipse shape acts exactly like an obox. Its string representation is the same as the string representation of an obox that bounds the ellipse.

point

A point's string representation is (x, y) where x and y are the coordinates of the point.

circle

A circle is defined by its center and radius. The string representation is $(x, y)r$ where x and y are the center point's coordinates, and r is the radius.

polygon

A polygon is an ordered list of points, with line segments connecting them. Closed polygons have a line segment connecting the first and the last point, while open polygons do not. The string representation for both is: (x_1, y_1), (x_2, y_2),..., (x_n, y_n), where x_1, y_1, x_2, y_2, ..., x_n, y_n are the coordinates of points of the polygon. Note that open polygons must have at least two points and closed polygons must have at least three. Currently, only closed polygons are supported.

References

[1] J.F. Allen and G. Ferguson. Actions and events in interval temporal logic. *Journal of Logic Computation*, 4:531–579, 1994.

[2] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *Int. J. Comput. Vision*, 12:42–77, 1994.

[3] J. Black and T. Ellis. Multi camera image tracking. *Image and Vision Computing*, 24(11):1256–1267, 2006.

[4] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *Proc. IEEE Int. Conf. Computer Vision*, pages 1395–1402, Oct. 2005.

[5] A.F. Bobick and A.D. Wilson. A state-based technique to the representation and recognition of gesture. *IEEE Trans. Pattern Anal. Machine Intell.*, 19:1325–1337, 1997.

[6] M. Brand. Understanding manipulation in video. In *IEEE Int. Conf. Automatic Face and Gesture Recognition*, pages 94–99, Oct. 1996.

[7] M. Brand and V. Kettnaker. Discovery and segmentation of activities in video. *IEEE Trans. Pattern Anal. Machine Intell.*, 22:844–851, 2000.

[8] C. Calvo, A. Micarelli, and E. Sangineto. Automatic annotation of tennis video sequences. In *Proc. DAGM Symposium on Pattern Recognition*, volume LNCS 2449, pages 540–547. Springer, Sept. 2002.

[9] Y. Caspi and M. Irani. Spatio-temporal alignment of sequences. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24:1409–1424, 2002.

[10] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 1932 – 1939, June 2009.

[11] R.T. Collins, A.J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, and O. Hasegawa. A system for video surveillance and monitoring. Technical report, 2000.

[12] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. IEEE *Trans. Pattern Analysis and Machine Intelligence*, 25:564–577, 2003.

[13] I. Cox and S. Higorani. An efficient implementation of reid's mht algorithm and its evaluation for the purpose of visual tracking. *IEEE Trans. Pattern analysis and Machine Intelligence*, 18(2):138–150, 1996.

[14] R. Dahyot, A. Kokaram, N. Rea, and H. Denman. Joint audio visual retrieval for tennis broadcasts. In *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, pages 561–564. IEEE, April 2003.

[15] J. Martinez del Rincon, J E. Herrero-Jaraba, J.R. Gomez, and C. Orrite-Urunuela. Automatic left luggage detection and tracking using multi-camera. In *Proc. IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 59–66, June 2006.

[16] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern classification.* Wiley, 2000.

[17] A. Ekin, A. M. Tekalp, and R. Mehrotra. Automatic soccer video analysis and summarization. *IEEE Trans. Circuits Systems for Video Technology*, 12:796–807, 2003.

[18] D. Farin, J. Han, and P.H.N. de With. Fast camera calibration for the analysis of sports sequences. In *IEEE Proc. ICME*, July 2005.

[19] N. Friedman and S. Russell. Image segmentation in video sequences: a probabilistic approach. In *Proc. 13th Conf. Uncertainty in Artificial Intelligence*, pages 175–181, Aug. 1997.

[20] H. Fujiyoshi, A. Lipton, and T. Kanade. Real-time human motion analysis by image skeletonization. *IEICE Trans. Information and System*, 87:113–120, 2004.

[21] D.M. Gavrila and S. Munder. Multi-cue pedestrian detection and tracking from a moving vehicle. *Internal Journal of Computer Vision*, 73(1):41–59, June 2007.

[22] M. Gelgon, P. Bouthemy, and J.P. Le-Cadre. Recovery of the trajectories of multiple moving objects in an image sequence with a pmht approach. *Journal of Image and Vision Computing*, 23(1):19–31, 2005.

[23] W. Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in a site. In *Proc. IEEE Conf. Computure Vision and Pattern Recognition*, pages 22–29, June 1998.

[24] T. Tuytelaars G.Willems, J.H. Becker. Exemplar-based action recognition in video. In *Proc. British Machine Vision Conference*, pages 1–8, Aug. 2009.

[25] J. Han and P.H.N. de With. Real-time multiple people tracking for automatic group-behavior evaluation in delivery simulation training. *Springer Multimedia Tools Applications*, 51:913–933, 2010.

[26] J. Han, D. Farin, and P.H.N. de With. Multi-level analysis of sports video sequences. In *Proc. Multimedia Content Analysis Management and Retrieval*, pages 6073:1–12. SPIE, Jan. 2007.

[27] J. Han, D. Farin, and P.H.N. de With. A real-time augmented-reality system for sports broadcast video enhancement. In *Proc. Multimedia*, pages 337–340. ACM, Sept. 2007.

[28] J. Han, D. Farin, P.H.N. de With, and W. Lao. Automatic tracking method for sports video analysis. In *Proc. Symposium on Information Theory in the Benelux*, pages 309–316. IEEE, May 2005.

[29] J. Han, D. Farin, P.H.N. de With, and W. Lao. An automatic analyzer for sports video databases using visual cues and real-world modeling. In *Proc. Int. Conf. Consumer Electronics*, pages 477–478. IEEE, Jan. 2006.

[30] J. Han, D. Farin, P.H.N. de With, and W. Lao. Real-time video content analysis tool for consumer media storage system. *IEEE Transactions on Consumer Electronics*, 52:870–878, 2006.

[31] J. Han, M. Feng, and P.H.N. de With. A real-time video surveillance system with human occlusion handling using nonlinear regression. In *Proc. Int. Multimedia & Expo*, pages 305–308. IEEE, July 2008.

[32] J. Han, W. Lao, and P.H.N. de With. Scene-level analysis for tennis sports video using weighted linear combination of visual cues. In *Proc. Int. Conf. Internet and Multimedia Systems and Applications*, pages 193–197, Feb. 2006.

[33] I. Haritaoglu, D. Harwood, and L. Davis. W4: real-time surveillance of people and their activities. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22:809–830, 2000.

[34] R. Hartley. *Multiple view geometry in computer vision.* Cambridge University Press, 2000.

[35] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang. Face recognition using laplacianfaces. *Transactions on Pattern Analysis and Machine Intelligence*, 27:328–340, 2005.

[36] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *Trans. Systems, Man and Cybernetics*, 34:334–3526, 2004.

[37] Y.A. Ivanov and A.F. Boblic. Recognition of visual activities and interactions by stochastic parsing. *Computer Vision and Image Understanding*, 22:852–872, 2000.

[38] O. Javed, Z. Rasheed, K. Shafique, and M. Shah. Tracking across multiple cameras with disjoint views. In *Proc. IEEE Int. Conf. Computer Vision*, volume 2, pages 952–957, Oct. 2003.

[39] K. Jia and D. Yeung. Human action recognition using local spatio-temporal discriminant embedding. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, pages 1–8, Dec. 2008.

[40] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. *Image Vis. Comput.*, 14:609–615, 1996.

[41] E. Kijak, L. Oisel, and P. Gros. Temporal structure analysis of broadcast tennis video using hidden markov models. In *Proc. Storage and Retrieval for Media Databases*, pages 289–299. SPIE, Jan. 2003.

[42] V. Kobla and D. Doermann. Detection of slow-motion replays for identify sports videos. In *3rd Workshop on Multimedia Signal Processing*, pages 135–140. IEEE, Sept. 1999.

[43] A. Kojima, T. Tamura, and K. Fukunaga. Natural language description of human activities from video images based on concept hierarchy of actions. *IEEE Int. Journal of Computer Vision*, 50(2):171–184, 2002.

[44] W. Lao, J. Han, and P.H.N. de With. Automatic sports video analysis using audio clues and context knowledge. In *Proc. Int. Conf. Internet and Multimedia Systems and Applications*, pages 198–202, Feb. 2006.

[45] W. Lao, J. Han, and P.H.N. de With. Ball-path inference based on a combination of audio and video clues in tennis video sequences. In *Proc. Benelux/DSP Valley Signal Processing Symposium*. IEEE, March 2006.

[46] W. Lao, J. Han, and P.H.N. de With. A matching-based approach for human motion analysis. In *Proc. Int. Conf. Multimedia Modeling*, volume LNCS 4352, pages 405–414. Springer, Jan. 2007.

[47] W. Lao, J. Han, and P.H.N. de With. Multi-module human motion analysis from a monocular video. In *Proc. Electronic Imaging*, pages 65060M.1–65060M.9, Jan. 2007.

[48] W. Lao, J. Han, and P.H.N. de With. Fast detection and modeling of human-body parts from monocular video. In *Proc. Int. Conf. Articulated Motion and Deformable Objects*, pages 380–389. Springer, July 2008.

[49] W. Lao, J. Han, and P.H.N. de With. Automatic video-based human motion analyzer for consumer surveillance system. *IEEE Trans. Consumer Electronics*, 55:591–598, 2009.

[50] W. Lao, J. Han, and P.H.N. de With. Fast detection and modeling of human-body parts from monocular video. In *Proc. IEEE Int. Conf. Consumer Electronic*, pages 1–2, Jan. 2009.

[51] W. Lao, J. Han, and P.H.N. de With. Multi-level human motion analysis for surveillance applications. In *Proc. SPIE Visual Communications and Image Processing*, pages 1–9, Jan. 2009.

[52] W. Lao, J. Han, and P.H.N. de With. Flexible human behavior analysis framework for video surveillance applications. *Hindawi International Journal of Digital Multimedia Broadcasting*, 2010:920121:1–9, 2010.

[53] H.J. Lee and Z. Chen. Knowledge-guided visual perception of 3d human gait from a single image sequence. *Trans. Systems, Man, Cybernetics*, 22:336–342, 2002.

[54] M.K. Leung and Y.H. Yang. First sight: a human body outline labeling system. *Trans. Pattern Analysis and Machine Intelligence*, 17:359–377, 1995.

[55] H. Li, S. Wu, S. Ba, S. Lin, and Y. Zhang. Automatic detection and recognition of athlete actions in diving video. In *Proc. Int. Conf. Multimedia Modeling*, volume LNCS 4352, pages 73–82. Springer, Jan. 2007.

[56] A.J. Lipton, H. Fujiyoshi, and R. S. Patil. Moving target classification and tracking from real-time video. In *IEEE Workshop Applications of Computer Vision*, pages 8–14, Oct. 1998.

[57] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[58] V.Y. Mariano, J. Min, J.H. Park, R. Kasturi, D. Mihalcik, D. Doermann, and T. Drayer. Performance evaluationof object detection algorithms. In *Proc. IEEE Int. Conf. Pattern Recognition*, pages 965–968, Aug. 2002.

[59] S. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler. Tracking groups of people. *Comput. Vis. Image Understanding*, 80(1):42–56, 2000.

[60] S.J. McKenna, S. Jabri, Z. Duric, and H. Wechsler. Tracking interacting people. In *Int. Conf. Automatic Face and Gesture Recognition*, pages 348–353. IEEE, March 2000.

[61] D. Meyer, J. Psl, and H. Niemann. Gait classification with hmms for trajectories of body parts extracted by mixture densities. In *British Machine Vision Conf.*, pages 459–468, 1998.

[62] A. Mittal and L. Davis. M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *International Journal of Computer Vision*, 51(3):189–203, 2003.

[63] H. Miyamori. Video annotation for content-based retrieval using human behavior analysis and domain knowledge. In *Proc. Int. Conf. Automatic Face and Gesture Recognition*, pages 320–325. IEEE, March 2000.

[64] T. B. Moeslund, A. Hilton, and V.Kruger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104:90–126, 2006.

[65] T.B. Moeslund and E. Granum. Tracking and modeling people in video sequences. *Computer Vision and Image Understanding*, 81:285–302, 2001.

[66] T.B. Moeslund and E. Granum. Human body model acquisition and tracking using voxel data. *Int. Journal of Computer Vision*, 53:199–223, 2003.

[67] G. Mori and J. Malik. Recovering 3d human body configurations using shape contexts. *IEEE Trans. Pattern Anal. Machine Intell.*, 28(7):1052–1062.

[68] G. Mori and J. Malik. Recovering 3d human body configurations using shape contexts. *IEEE Trans. Pattern Anal. Machine Intell.*, 28:1052–1062, 2006.

[69] K. Nummiaro, E. Koller-Meier, and L.V. Gool. A color-based particle filter. In *Proc. IEEE Int. Workshop on Generative Model-based Vision*, pages 53–60, June 2002.

[70] N.M. Oliver, B. Rosario, and A.P. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Trans. Pattern Anal. Machine Intell.*, 22:831–843, 2000.

[71] M. Ozuysal, V. Lepetit, F. Fleuret, and P. Fua. Feature harvesting for tracking-by-detection. In *Proc. European Conference on Computer Vision*, volume LNCS 3953, pages 592–605. Springer, May 2006.

[72] H. Pan, P. Beek, and M. Sezan. Detection of slow-motion replay segments in sports video for highlights generation. In *ICASSP*, pages 1649–1652. IEEE, May 2001.

[73] S. Park and K. Aggarwal. Simultaneous tracking of multiple body parts of interacting persons. *Computer Vision and Image Understanding*, 102:1–21, 2006.

[74] N. Peterfreund. Robust tracking of position and velocity with kalman snakes. *IEEE Trans. Pattern Anal. Machine Intell.*, 22:564–569, 2000.

[75] P. Peursum, H. Bui, S. Venkatesh, and G. West. Computer vision system for in-house video surveillance. *IEE proc. Vision, Image and Signal Processing*, 152:242–249, 2005.

[76] P. Peursum, H. Bui, S. Venkatesh, and G. West. Robust recognition and segmentation of human actions using hmms with missing observations. *EURASIP Journal on Applied Signal Processing*, 13:2110–2126, 2005.

[77] K. Philippe. The 4+1 view model of architecture. *IEEE Software*, 12:42–50, 1995.

[78] J. Piater and J. Crowley. Multi-model tracking of interacting targets using gaussian approximations. In *Proc. IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance*, pages 141–147, Dec. 2001.

[79] A. Pikrakis, T. Giannakopoulos, and S. Theodoridis. Gunshot detection in audio streams from movies by means of dynamic programming and bayesian networks. In *Proc. IEEE Int. Conf. Acoustic, Signal and Signal Processing*, pages 21–24, Apr. 2008.

[80] G. Pingali, Y. Jean, and I. Carlbom. Real time tracking for enhanced tennis broadcasts. In *Proc. Computer Vision and Pattern Recognition*, pages 260–265. IEEE, June 1998.

[81] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE*, 77:257–285, 1998.

[82] C. Rao, A. Yilmaz, and M. Shah. View-invariant representation and recognition of actions. *International Journal of Computer Vision*, 50:203–226, 2002.

[83] N. Rea, R. Dahyot, and A. Kokaram. Classification and representation of semantic content in broadcast tennis videos. In *Proc. ICIP*, pages 1204–1207. IEEE, Sept. 2005.

[84] P. Remagnino, T. Tan, and K. Baker. Agent orientated annotation in model based visual surveillance. In *Proc. IEEE Int. Conf. Computer Vision*, pages 857–862, Jan. 1998.

[85] M.S. Ryoo and J.K. Aggarwal. Recognition of composite human activities through context-free grammar based representation. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 1709–1718. IEEE, June 2006.

[86] L. Shao and X. Chen. Histogram of body poses and spectral regression. In *Proc. British Machine Vision Conference*, pages 1–8, Aug. 2010.

[87] L. Shao, L. Ji, Y. Liu, and J. Zhang. Human action segmentation and recognition via motion and shape analysis. *Pattern Recognition Letters*, doi:10.1016/j.patrec, 2011.

[88] J. Shi and R. Gross. The cmu motion of body (mobo) database. Technical report, 2001.

[89] N. Siebel and S. Maybank. Fusion of multiple tracking algorithms for robust people tracking. In *Proc. European Conference on Computer Vision*, pages 373–387. Springer, May 2002.

[90] M. Singh, A. Basu, and M. Mandal. Human activity recognition based on silhouette directionality. *IEEE Trans. Circuits and Systems for Video Technology*, 18(9):1280–1292, 2008.

[91] P. Smith, M. Shah, and N. V. Lobo. Integrating multiple levels of zoom to enable activity recognition. *Computer Vision and Image Understanding*, 36(3):33–51, 2005.

[92] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Int. Conf. Computer Vision and Pattern Recognition*, pages 246–252. IEEE, June 1999.

[93] G. Sudhir, C. Lee, and K. Jain. Automatic classification of tennis video for high-level content-based retrieval. In *Proc. Int. Workshop on Content Based Access of Image and Video Databases*, pages 81–90. IEEE, Jan. 1998.

[94] N. Sumpter and A. Bulpitt. Learning spatio-temporal patterns for predicting object behavior. *Image Vis. Comput.*, 18:697–704, 2000.

[95] H.Z. Sun, T. Feng, and T. N. Tan. Robust extraction of moving objects from image sequences. In *Proc. Asian Conf. Computer Vision*, pages 961–964. Springer, Jan. 1998.

[96] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: principles and practice of background maintenance. In *Proc. Int. Conf. Computer Vision*, pages 255–261, Sept. 1999.

[97] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proc. Int. Conf. Computer Vision*, pages 734–741, Oct. 2003.

[98] K. Wan, X. Yan, X. Yu, and C. Xu. Real-time goal-mouth detection in mpeg soccer video. In *Proc. Multimedia*, pages 311–314. ACM, Nov. 2003.

[99] Y. Wang, Z. Liu, and J-C Huang. Multimedia content analysis using both audio and visual clues. *IEEE Trans. Circuits Systems for Video Technology*, 17:12–36, 2000.

[100] M. Xin, L-Y Duan, C-S Xu, and Q. Tian. A fusion scheme of visual and auditory modalities for event detection in sports video. In *Proc. Int. Conf. Multimedia & Expo, 2003*, pages 333–336. IEEE, July 2003.

[101] Z. Xiong, R. Radhakrishnan, A. Divakaran, and T. S. Huang. Audio events detection based highlights extraction from baseball, golf and soccer games in a unified framework. In *Proc. Int. Conf. Multimedia & Expo*, pages 401–404. IEEE, July 2003.

[102] C. Yu, J. Hwang, G. Ho, and C. Hsieh. Automatic human body tracking and modeling from monocular video sequences. In *IEEE Proc. Int. Conf. Acoustics, Speech and Signal Processing*, volume 1, pages 917–920, May 2007.

[103] X. Yu, C-H. Sim, J. R. Wang, and L. F Cheong. A trajectory-based ball detection and tracking algorithm in broadcast tennis video. In *Proc. ICIP*, pages 1049–1052. IEEE, July 2004.

[104] L. Zelnik-Manor and M. Irani. Event-based analysis of video. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 123–130. IEEE, Dec. 2001.

[105] T. Zhao and R. Nevatia. Tracking multiple humans in crowded environment. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, pages 406–413, June 2004.

[106] F. Zheng, L. Shao, and Z. Song. Eigen-space learning using semi-supervised diffusion maps for human action recognition. In *Proc. ACM Int. Conf. Image and Video Retrieval*, pages 1–8, Jul. 2010.

[107] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 819–826. IEEE, June 2004.

[108] Z. Zhou, Y-C. Chung, Z. He, X. Han, and M. Keller. Activity analysis, summarization, and visualization for indoor human activity monitoring. *IEEE Trans. Circuits and Systems for Video Technology*, 18:1489–1498, 2008.

[109] Z. Zhou, A. Prugel-Bennett, and R. Damper. A bayesian framework for extracting human gait using strong prior knowledge. *IEEE Trans. Pattern Anal. Machine Intell.*, 28:1738–1752, 2006.

[110] 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.

Acknowledgements

Seven years ago, I was very delighted to have an interview with prof.dr. Peter de With in Singapore and afterwards obtained the offer to work as a Ph.D. candidate at the VCA group at Eindhoven University of Technology. This opportunity enables me to land on Europe and turns on my new pages in both academic exploration and personal experiences.

First of all, I wish to sincerely thank my promoter prof.dr. Peter de With for his guidance, encouragement, support, patience, persistence and enthusiasm during all these years. His advices, ideas and suggestions on my research and thesis writing are invaluable. This thesis work would have never been accomplished without Peter's arrangement and support. Whenever I consulted with him confused, I would afterwards become enlightened, inspired, and enthusiastic. Peter always impresses me about his endless energy and firm confidence.

I would also like to extend my appreciation to my co-promoter and friend dr. Jungong Han. Without his daily supervision and kind support, it is very challenging to go through the rigorous road towards a Ph.D. I will always remember our joyful conversations at the lunch table on topics from scientific research to life experience.

I would like to express my gratitude to the promotion committee for their efforts in reading the thesis thoroughly and providing very valuable comments.

Special thanks to Yueyue for her willing to design my thesis cover, and Rob, Lykele for their kind help in translating the thesis summary into a Dutch version. I am grateful to Anja for her patient and timely assistances in undertaking my graduation procedure.

The research work presented in this thesis has been carried out in cooperation with several international and local partners within the ITEA project Candela and Cantata. I would like to thank dr. Egbert Jaspers from ViNotion for his management and support in these projects.

Thanks to all my colleagues at the VCA group for their support during the period of my thesis work. Sincerely I would also like to thank my friends at ACSSE and PromoVE, who have, at every step of the way, supported me in the pursuit of my Ph.D. degree.

Last but not the least, I would like to extend my arms and give a big hug to my family. I thank my parents for their love in all aspects of my life. Without their love, it is impossible for me to grow up and make progress ever since. Without the unlimited support from my family, my development would never have reached this level. Finally, loving kiss to my dear wife Yingying for her endless encouragements, silent sacrifice, precious love and bringing me a lovely daughter.

Weilun Lao
Guangzhou,
September 2011

Curriculum Vitae

Weilun Lao obtained his Bachelor's degree in Dept. Automation from the South China University of Technology (SCUT), China in 2002. From 2003 to 2005, he conducted research work in Institute for Infocomm Research (I^2R), Singapore and received his Master's degree in Dept. Electrical & Computer Engineering at the National University of Singapore (NUS). From 2005 to 2009, he was a Ph.D. student in the Video Coding and Architectures (VCA) group of the Electrical Engineering department at Eindhoven University of Technology (TU/e), Eindhoven, The Netherlands. His research interests include multimedia content analysis, 3-D human motion analysis and computer vision. He served as a committee member for PromoVE (Ph.D. students association) at TU/e and obtained an award for Outstanding Chinese Overseas Students in 2007. He is currently active as a software designer for establishing reliable transaction communication software in databases.