Vision-based 3D Human Motion Analysis in a Hierarchical Way

Proefschrift

ter verkrijging van de graad van doctor aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op donderdag 27 juni 2013 om 15.00 uur
door

Feifei HUO

Master of Engineering, Xidian University, China
Master of Science, Delft University of Technology
geboren te Luoyang, Henan Province, China
Dit proefschrift is goedgekeurd door de promotor:
Prof. dr. ir. M.J.T. Reinders

Copromotor:
Dr. E.A. Hendriks

Samenstelling promotiecommissie:

Rector Magnificus, voorzitter
Prof. dr. ir. M.J.T. Reinders, Technische Universiteit Delft, promotor
Dr. E.A. Hendriks, Technische Universiteit Delft, copromotor
Prof. dr. ir. P.P. Jonker, Technische Universiteit Delft
Prof. dr. C.M. Jonker, Technische Universiteit Delft
Prof. dr. R.C. Veltkamp, Universiteit Utrecht
Prof. dr. ir. B.J.A. Krose, Universiteit van Amsterdam
Prof. dr. E.O. Postma, Universiteit van Tilburg
Prof. dr. E. Eisemann, Technische Universiteit Delft, reservelid

This work was supported by the GATE (Game research for Training and Entertainment) project.

This work was carried out in the ASCI graduate school.
ASCI dissertation series number 283.

ISBN 978-94-9153-609-0
Copyright © 2013 by F. Huo
All right reserved. No part of this thesis may be produced or transmitted in any form or by any means, electronic, mechanical, photocopying, any information storage or retrieval system, or otherwise, without written permission from the copyright owner.
to my family
# CONTENTS

1 **Introduction** .................................................................................................................. 1  
  1.1 Vision-based human motion analysis ............................................................................. 2  
    1.1.1 Human motion tracking: efficiency and simplicity .................................................. 2  
    1.1.2 Pose estimation: occlusion handling ......................................................................... 2  
    1.1.3 Activity recognition: interaction recognition ......................................................... 3  
  1.2 Related work .................................................................................................................. 3  
  1.3 The general approach .................................................................................................. 6  
  1.4 Thesis outline ............................................................................................................... 8  

2 **Markerless Human Motion Capture and Pose Recognition** .............................................. 9  
  2.1 Introduction .................................................................................................................. 10  
  2.2 Previous research ......................................................................................................... 10  
  2.3 Methodology ............................................................................................................... 11  
    2.3.1 Background subtraction .......................................................................................... 11  
    2.3.2 2D model for human torso detection and tracking ................................................. 12  
    2.3.3 Hand detection and tracking .................................................................................. 13  
    2.3.4 3D reconstruction .................................................................................................. 14  
    2.3.5 Feature space construction ..................................................................................... 14  
  2.4 Experiments .................................................................................................................. 15  
    2.4.1 Video recording ...................................................................................................... 15  
    2.4.2 Implementation ...................................................................................................... 15  
    2.4.3 Pose classification .................................................................................................. 15  
  2.5 Results and discussions ............................................................................................... 16  
  2.6 Spatial game application ............................................................................................... 18  
    2.6.1 Implementation ...................................................................................................... 18  
    2.6.2 Discussions .......................................................................................................... 19  
  2.7 Conclusions ................................................................................................................. 20  

3 **Real Time Multiple People Tracking and Pose Estimation** ............................................ 21  
  3.1 Introduction .................................................................................................................. 22  
  3.2 Related work ................................................................................................................ 22  
  3.3 Methodology ............................................................................................................... 22  
    3.3.1 Initialization step .................................................................................................... 25  
    3.3.2 Appearance model building ................................................................................... 26
Chapter 1

Introduction

Nowadays, fast and cheap computer hardware is available. Combined with increasingly cheaper and better digital cameras, computer vision based applications have become more and more widespread [1, 58, 59]. One of the key components of realizing these applications is computer vision based image or video understanding. But what is “computer vision” exactly? Is it that a computer sees what we see? Apparently not; a camera already can do it. Is it that a computer helps us to see better? No, that is also not difficult; image processing, e.g. noise removal or sharpening, does a good job. Is it that a computer can understand image/video as humans do? From our point of view, this is the closest answer we would choose. “Let a computer understand image/video just as humans do”, is a hard issue and is what computer vision researchers have been trying to do in the last two decades.

Computer vision finds its way in many fields, such as face detection [2, 3], facial expression recognition [4-6], gesture recognition [7-9], pedestrian detection and tracking [10-12], human pose estimation [13-15]. Recently, vision-based human motion tracking has drawn more and more attention [16, 17, 57]. For instance, due to the increase of public surveillance cameras, a huge amount of video data is generated everyday and everywhere in the world. It is expensive, boring, and time consuming for humans sitting in front of monitors and analyzing all these video data. Consequently, computer vision based methods are developed and used for surveillance tasks instead of humans.

In the early stages of computer vision, researches on human motion tracking are addressing a single person and to a fixed position in front of a camera. In the last decade, attention has been paid to multiple people tracking. The ability to robustly track multiple people in a room opens new applications. Examples of such applications are pose-driven spatial games [60], in which players get rid of controllers and play games using intuitive body movements and poses, and serious games in which people interact with a simulated virtual world in real-time. In the health-care sector, e.g.: the search for objective measures of surgical skill can be supported by a vision system tracking a surgeon's movements [133]; nursing quality can be improved by monitoring nursing care [52]; or, emergency alarm can be triggered by analyzing patients’ behavior.

The traditional way of solving multiple people motion tracking problems starts from single person motion tracking. Moreover, multiple people motion tracking is obviously more challenging compared with single person motion tracking as people may occlude each other. It is difficult to handle large occlusions efficiently and accurately. The problem of multiple
people body-part detection and tracking is still an unsolved problem and researchers are actively working in this area [29, 31, 32, 34, 35, 44].

The objectives of this PhD thesis were to develop fast and robust algorithms that can detect, track, and model accurately and robustly individual persons in the real 3D world, and to recognize motion of individuals, identify the interaction between persons, and to design applications where interaction between humans and computers is required [60].

1.1 Vision-based human motion analysis

Vision-based human motion analysis is a broad topic by itself. In this section, we briefly explain the three main areas that are most important with regards to the research described in this thesis. The three main research areas are: human motion tracking, pose estimation, and activity recognition.

1.1.1 Human motion tracking: efficiency and simplicity

Human motion tracking is a term used to describe the process of recording movement of persons [61]. One way of human motion tracking is marker-based. People need to wear specific suits with markers on them to track the movement of different body parts, which is not convenient for real applications. To avoid this inconveniency, markerless human motion tracking approaches are desired. Markerless approaches simply use cameras to record body movements and people are totally free of any intrusive sensors. There have been researches carried out on markerless human motion tracking [18-22, 30-37, 46]. Previous tracking algorithms, for example the Kalman filter [23, 24], have been limited in the range of probability distributions they represent. Probabilistic algorithms, such as the particle filtering [25, 26, 45], have been developed which allow general representations of probability distributions. Particle filtering based approaches make human motion tracking robust and error tolerant. These approaches and derived methods [27-29], which are designed to cope with some degree of uncertainty, accurately predict the likelihood of body movements.

Most of the probabilistic approaches for body tracking can already achieve high tracking accuracy. However, due to the complexity of these approaches, the computational time is far away from real-time. In order to achieve real-time performance, efficiency and simplicity of the algorithm need to be taken into account. The most time consuming part of body tracking lies in the search of possible body configurations. Therefore, reducing the search space dimensionality is crucial for realizing real-time human motion tracking. In this thesis, we put effort in investigating how to reduce the search space dimensionality in order to achieve real-time tracking.

1.1.2 Pose estimation: occlusion handling

Pose estimation is referred to as the process of estimating a person’s pose. For multiple people pose estimation, identical estimators can be used to estimate each individual’s pose simultaneously. The main research question is whether estimating multiple people’s poses can work well when occlusion occurs. By occlusion, we mean that in a 2D image, one person
is occluded by another person (inter-person occlusion); or one body part is occluded by another body part (self occlusion). Occlusion is a difficult problem to deal with, since it is hard to predict people’s behavior. For instance, people may change their moving directions after occlusion or still move in the same direction. Apart from this, the appearance of people may also change, such as from frontal view to lateral view. Moreover, when people are close to each other, it is easy to cause shadows on each other’s body. All of this makes it a challenge to cope with occlusions of people in a multiple people pose estimation system. Some existing methods aim at dealing with self occlusions, or with minor occlusions between individuals [32, 35]. They are not suitable for handling severe occlusions between individuals. By severe occlusions, we mean that more than 50% of the body part is occluded, which is common in indoor or outdoor environments. In order to deal with severe occlusions, multiple views and various image features are necessary for building and analyzing the identity (descriptor) of each individual and each body part. The research question is how to extract reliable image features for body parts detection and how to combine multiple views in an efficient way.

1.1.3 Activity recognition: interaction recognition

Being able to estimate multiple people’s poses, our work continues with human interaction recognition. The challenge of human interaction recognition is to construct feature spaces to represent different interactions, such as two-person interactions [49, 54, 70, 73], group activities [48, 50, 52, 74-76], and social interactions [77]. Since recognition performance depends on reliable feature extraction, some researches focus on extracting features and building classifiers to achieve the best classification performance. Little attention has been paid on investigating what are the most informative features. For example, should you use spatial features, or temporal features, or a combination of both to represent certain types of interaction? In this thesis, we focus on analyzing the influence of different types of features on classifier performances.

1.2 Related work

The number of papers on human motion analysis has grown. Overviews of vision-based human motion analysis methods can be found in [16, 17, 57]. Based on the focus of this thesis, we divide the prior work into three parts: human motion tracking, pose estimation, and activity recognition.

Human motion tracking can be divided into two categories: monocular approaches [30-32, 39] and multi-view approaches [27, 29, 33-36, 62]. Monocular approaches use the video data from a single camera to perform tracking. Some of the frontier approaches are “Pfinder: person finder” [30] and “W4: what, where, when, and who” [31]. Pfinder is a real-time system for tracking people and interpreting their behaviors. It has been used in several applications, such as video games, gesture recognition, interactive interface, virtual reality. Pfinder uses simple 2D models for detection and tracking of human body. Different body parts are labeled and located through contour shape analysis. It is shown that tracking performance is significantly increased by taking color into account. However Pfinder
describes the human body with a blob model. Multiple users cause problems in certain applications, e.g. gesture recognition, because Pfinder attempts to interpret detected blobs as one distinct human figure. W4 is a real-time visual system for detecting and tracking multiple people in an outdoor environment. It is capable of tracking multiple people simultaneously even with occlusions. A dynamic appearance model is constructed for each person by combining gray-scale textual appearance and shape information. The appearance model is used to identify people after an occlusion happened. W4 is primarily designed for outdoor surveillance. Since only the silhouette is used to estimate body postures, only distinct postures: standing, sitting, crawling, and lying can be determined, not more subtle body poses. Both Pfinder and W4 achieve real-time performance, which increased their applicability in surveillance and human computer interaction (HCI). More recently, color-based appearance models are favored in multiple people tracking due to its effectiveness in occlusion handling. In [39] a complete automatic system is developed to track people by learning their appearance. Their tracker is able to track people from a single view with automatic initialization and with a complex background. The system was tested on frames from commercial and unscripted videos. One major limitation of this system is, however, how it handles occlusions. When inter-person occlusions happen in the scene, tracking of the occluded person will fail temporarily. In [32], a data-driven Markov chain Monte Carlo (DD-MCMC) approach is proposed to estimate the full 3D body poses. Partial occlusion between persons is handled with the appearance model of each person. The approach is able to initialize the tracker automatically and recover from partial tracking failure due to occlusion. However, the total processing time for each frame took an average of 5 minutes.

Due to the limitation of a single view, monocular approaches only can deal with self occlusions or inter-person occlusions to a limited extent. In order to better handle occlusions and estimate 3D human body pose, multi-view approaches have been proposed [27, 29, 34-36, 62].

Multi-view approaches utilize video sequences recorded from multiple cameras to detect and track people. In general, multi-view approaches are more effective at occlusion handling and at obtaining the precise 3D positions of tracked persons, compared with monocular approaches. Spfinder is an extension of Pfinder [62]. It uses a stereo system for recovering 3D description of humans in real time. Spfinder has been used in a small desk-area environment to capture 3D head and hands movements. Essentially the same 2D techniques are used in Spfinder as Pfinder to produce blob features. Therefore, Spfinder shares the same limitation as Pfinder in multiple people tracking. In [34] multiple people tracking is reliably achieved by processing each individual’s trajectory separately. A heuristic is used to find optimal trajectories over time. The system can track up to 6 persons in an indoor environment with 4 cameras, in spite of significant occlusions. Since the global optimization of trajectories is carried over 100-frame batches, there is a 4 second (given frame rate by 25 f/s) delay between image acquisition and results output. The system does not address pose estimation. In [27], three synchronized cameras are used to capture the motion of a single person. The proposed algorithm, termed annealed particle filtering, is capable of recovering full articulated body movement efficiently. Compared with standard Condensation [63], the computation time is reduced by over a factor of 10, which makes a considerable step towards real-time tracking of a single person. In [29], particle filters are used for multiple people pose estimation. In order to reduce the computational complexity, a hierarchical stochastic sampling scheme is proposed, which is more efficient compared with alternatives such as
1.2. Related work

annealed particle filtering and partitioned sampling. The proposed scheme is capable of tracking 2 persons using 5 cameras, with a processing time of 15 seconds per frame. In [35], an approach is proposed for automatic initialization and tracking of human poses. The approach combines bottom-up evidence with top-down priors to get efficient pose estimation. The proposed approach can handle self occlusions by using an appearance model and an occlusion map. However, the runtime is around 45 seconds per frame due to the search of possible part configurations. In [36] a framework is presented for 3D human upper body pose estimation in a complex environment. The framework integrates three components: single-frame pose recovery, temporal integration and model texture adaptation. The framework is able to automatically reinitialize after a period of failure. Although the proposed approach achieves high accuracy with large and challenging real-world outdoor data, the processing speed is slow.

In summary, most of the multi-view approaches can obtain high tracking accuracy. However, the complexity of these approaches dramatically hinders the online applications of these tracking systems. The most time consuming part of body tracking lies in the search of possible body configurations. In order to achieve fast human motion tracking, reducing the search space dimensionality is necessary. In this thesis, we put effort in investigating algorithms which can be used to reduce the search space dimensionality of body tracking.

Pose estimation can be divided into two categories: model-free approaches and model-based approaches. Model-free approaches do not explicitly use a prior model of the relative configuration of body parts [30, 37, 38]. These approaches first detect individual body parts and then assemble them together into a configuration which best fits the observations. However, the search of possible part configurations is quite time consuming. Model-based approaches use an explicit model of a person’s kinematics, shape, appearance, etc. Model-based approaches have been widely used for body pose tracking because they provide a detailed description of a human body in 2D [39, 40] or 3D [27, 32, 35, 41-44]. A 2D model directly relates the model to image features, but depth information cannot be exploited due to view limitation. However, depth information is crucial for handling occlusions. In order to deal with large occlusions between persons, a 3D model is more suitable. 3D models used are skeleton models, cylinder models, mesh models. In general, 3D models give a better representation of the human body than 2D models resulting in a better accuracy.

In [39] the human body is modeled as a puppet of 2D rectangles. The system first builds a puppet model of each person by learning the person’s appearance and then tracks the person by detecting the learned puppet model in each frame. Self occlusions are considered in constructing the 2D model. In [40] a 2D upper-body model is used for pose detection over continuous sign language video sequences. In order to correct limb ambiguity, the 2D model takes proper account of self occlusions. The robustness of the proposed approach is shown on three challenging videos with continuously changing background. In [27] the human body is represented by a 3D articulated model. The model is based on a kinematic chain consisting of 17 parts and each limb is parameterized as a truncated cone with elliptical cross-sections. The model gives a compact representation of the human body and has the advantage of computational simplicity. In [35] a slightly different 3D human body is used which consists of 10 body parts. The torso is modeled as a cuboid, while other body parts are modeled as cylinders. A variety of constraints, such as kinematic constrains, appearance constrains, and collusion constrains, are incorporated in the model. The system achieves significant improvement in self occlusion handling compared to existing techniques. In [44] a complex
3D model is used to describe each person. This model is a 3D rigid shape composed of a bone skeleton and a triangle mesh surface. The whole body is segmented into 15 parts and the skeleton configuration of the body is estimated by a segmentation based approach. The system is able to simultaneously track two persons in close interaction with high accuracy.

One major issue of multiple people pose estimation lies in inter-person occlusion handling. In order to track accurately multiple people’s poses simultaneously, our research includes explicitly inter-person occlusion handling using multiple views. Multiple views are combined in such a way that we rely more on the features derived from reliable views and less on those from occluded views. In this way, we reduce the confusion caused by the occluded view and improve the tracking accuracy.

Recognizing human activities can be classified into two categories: single level approaches and hierarchical approaches as indicated in [64]. Single level approaches [54-56, 65-68] directly use the extracted features for human activity recognition without any intermediate processes. Such approaches have been successfully used in recognizing individual activities, such as walking, running, and jumping. In [65] a view-based approach is developed for human movement representation and recognition. Each action is represented by a vector image composed of a 2D motion-energy image (MEI) and a 2D motion-history image (MHI). Template matching is used to construct the recognition system. The system runs in real-time. In [66] the 2D motion of 13 body joints is used to represent an action. Since affine projection is used to obtain normalized trajectories of an action, the proposed algorithm can recognize actions from various view points. In [54] attention (head orientation) and the local spatial and temporal feature are used for two people interaction recognition. An initial set of linear support-vector-machine (SVM) classifiers is trained for four interactions: hand shakes, high fives, hugs and kisses.

Hierarchical approaches [47, 51, 53, 69-72] are more often used for representing and recognizing complex human activities and interactions. Hierarchical approaches describe human activities at multiple levels. For instance, the low level models individual body parts motion; the intermediate level models single person actions; and the high level models two-person interactions [69]. In [72], a layered hidden Markov model (LHMM) representation is used to model human activities in a hierarchical manner. The LHMMs are composed of a cascade of HMMs. It is feasible to train each level of the hierarchy independently. They demonstrated that the accuracy of LHMMs is significantly higher than that of single, standard HMMs, given the same amount of training data. Moreover, the proposed LHMMs are more robust to environment changes than HMMs in an office-awareness application.

For human activity recognition, attentions have been paid to recognition performance, such as accuracy, recognition rate, but not to investigating what are the most informative features to represent certain types of interaction. In this thesis, we compare different ways of representing spatial and temporal information for the purpose of interaction recognition, by analyzing classifier performance on different feature spaces.

1.3 The general approach

The general goal of this thesis is to analyze, interpret, and respond to the motion of groups of persons. The main objectives were:
1.3. The general approach

To develop fast and robust algorithms that can detect, track, and model accurately and robustly individual persons in the real 3D world.

2. To recognize poses and motion of individuals, identify the interaction between persons, and to design applications where interaction between humans and computers is required.

To address these objectives, the general approach taken by us is illustrated in Figure 1.1. It shows three steps. In the first step, we detect and track individuals in a group of users in a complex environment with possibly difficult lighting conditions. In the second step, we estimate the upper-body pose of each individual with self occlusions and inter-person occlusions. The estimated 2D and 3D poses are further used for pose and human interaction recognition. In the last step, meaningful poses and interactions are represented and recognized using 2D and 3D joints positions. These three steps are hierarchically related, which allows us to understand video content from a low-level to a high-level. The main contribution of our work is that we proposed fast vision-based solutions for multiple people tracking problems. We focused on the simplicity and efficiency of the algorithm and achieved the processing speed of 10 to 13 frames per second. The proposed approaches are suitable in real applications, such as health care, education, training, serious games, etc.
1.4 Thesis outline

Chapter 2 and Chapter 3 describe the 2D approaches for people detection, tracking and pose estimation. We start with single person motion tracking and pose recognition. By combining the tracking results from two synchronized camera views, we construct 3D poses of the tracked person. The simplicity and efficiency of the approach allow us to achieve real-time performance. Further on, we demonstrate an application based on the proposed approach, which is presented in Chapter 2 (Figure 1.1: low level analysis).

We extended the single person motion analysis approach into a multiple people detection and tracking system. In order to deal with occlusions, a combined probability estimation approach is proposed to detect and track multiple people for pose estimation at the same time. The simplicity of the features and the simplified model allow close to real time performance of the tracker (10 to 13 frames per second for upper body tracking). The proposed approach can deal with most of the inner-person occlusions, as well as certain self occlusions. It is faster than the existing methods with comparable accuracy. In Chapter 3, we describe the proposed multiple people tracking and pose estimation approach (Figure 1.1: low level and middle level analysis).

In Chapter 4 we extend the 2D approach to a 3D approach to overcome the view limitation of the 2D approach. A 3D upper-body model is projected onto multiple camera views and image evidences from these views are collected for the identification of each individual. A global occlusion reasoning scheme is proposed to deal with severe inter-person occlusions while a local occlusion scheme is developed to handle self occlusions. The combination of global and local occlusion estimation results in significant improvement in system performance regarding to the tracking accuracy. Moreover, a hierarchical way of search is used for upper body tracking, which reduces the computational complexity. The proposed 3D multiple view approach is given in Chapter 4 (Figure 1.1: middle level analysis).

Chapter 5 proposes a single level method for interaction recognition (Figure 1.1: high level analysis). The method directly uses tracked 3D joint positions of two interacting persons to recognize their interactions: shake hands, introduce, point, punch, wave and push. Both spatial and temporal features are used to represent each interaction. We put the emphasis on investigating which are the most informative features to distinguish these interactions while keeping good recognition performance.

Chapter 6 concludes the thesis and discusses possible future research directions.

Chapter 2-4 have been published and Chapter 5 is under review.
Chapter 2

Markerless Human Motion Capture and Pose Recognition

Abstract

In this paper, we present an approach to capture markerless human motion and recognize human poses. Different body parts such as the torso and the hands are segmented from the whole body and tracked over time. A 2D model is used for the torso detection and tracking, while a skin color model is utilized for the hands tracking. Moreover, 3D location of these body parts are calculated and further used for pose recognition. By transferring the 2D and 3D coordinates of the torso and both hands into normalized feature space, simple classifiers, such as the nearest mean classifier, are sufficient for recognizing predefined key poses. The experimental results show that the proposed approach can effectively detect and track the torso and both hands in video sequences. Meanwhile, the extracted feature points are used for pose recognition and give good classification results of the multi-class problem. The implementation of the proposed approach is simple, easy to realize, and suitable for real gaming applications.

This chapter has been published as:


2.1 Introduction

Nowadays, with the availability of faster and cheaper computer hardware, combined with cheaper and better digital cameras, video-based applications have become more and more widespread. A well-known video-based application is man-machine interaction, in which people can use their facial expressions, gestures and poses to control e.g. virtual actors or (serious) games. Human motion capture has received much attention due to such applications [16]. However, many of them are marker-based [78, 79]. People need to wear specific suits with markers on it to track the movement of different body parts, which is not convenient for real applications. To solve this problem, a markerless human motion capture system is desired. In this paper an approach to capture human motions without markers is presented and the extracted feature points are used for pose recognition.

2.2 Previous research

Although required for many natural applications such as pose recognition, there is still no generic solution to markerless motion capture. In [21] the skeleton points of a human are computed by using a silhouette model. Instead of calculating the 3D position of skeleton points, a topology of the human body structure is employed for limb labeling. The proposed method can deal with various viewpoints of a person, such as front, rear and profile. It also gives a proper limb labeling for unspecified human postures. However, they only use the graph topology matching to label different body parts, which has difficulty in dealing with the situation that the arms are merged with the torso. A real-time human motion analysis system is presented in [18], which combines a silhouette-based approach and a color-blob-based approach to get feature points. Although they realize real-time tracking by using a PC cluster to process images from six views, this implementation is still quite expensive for practical applications. In [20] the proposed algorithm uses 2D images for gesture recognition. The thresholded Radon transform coefficients are used to extract the most important local regions. One of the limitations of this algorithm is that it can not deal with self-occlusion of the human body.

In contrast to previous work, in this paper we introduce an effective method to track the movement of different body parts, such as torso and hands. The 3D location of these body parts are calculated and used for human pose recognition. The proposed human motion capture and pose recognition system is illustrated in Figure 2.1. The first step is human body detection and body parts segmentation, using multiple features, such as shape, contour and color. The second step is feature points representation and tracking in subsequent video frames. The 3D positions of selected feature points are calculated by using multiple calibrated cameras. The last step is pose recognition by using relative positions of selected feature points.
2.3 Methodology

2.3.1 Background subtraction

Motion is one of the important visual cues to find out the “interesting object” in the scene. Therefore, in our approach, we use background subtraction to segment moving objects. The background image is built by using a mixture of $k$ Gaussian models, which is presented in [80]. In order to deal with changing lighting conditions, the background image is updated over time by current frames. This method can also handle tracking of moving objects through cluttered scenes. An example of the obtained foreground binary image is shown in Figure 2.2 (a).
2.3.2 2D model for human torso detection and tracking

For the detection and tracking of humans, we applied a basic 2D model of a human’s head-shoulder-upperbody. This model is simple, but generic and successfully applied in [81]. The model is composed of two rectangles (Figure 2.2 (b)) and parameterized as \( p = (x, y, \text{scale}) \). \( x, y \) represent the position of the model in a 2D image and scale indicates the size of the model. Since a full searching is not feasible due to time constraints, and particle filters can deal with non-Gaussian motion models and multiple instances, we applied a particle filter both for people detection and tracking.

For people detection, an initial frame in a video sequence is used. The initial frame is chosen as the frame that shows a person with a specific pose (Figure 2.2 (b)). It indicates the start of the system. In order to reduce the search region and realize multiple people detection, the binary image of the initial frame is first segmented into connected blobs. Blobs which are impossible to include persons are discarded by judging their size. Then a particle filter is used on each candidate blob to determine if it includes a person or not. In particle filtering, particles are represented as \( p^{(n)} = (x^{(n)}, y^{(n)}, \text{scale}^{(n)}) \) (Figure 2.3 (a)). The position parameters \((x^{(n)}, y^{(n)})\) and scale parameters \((\text{scale}^{(n)})\) are initialized with a Gaussian distribution. If the coordinate of the upper left corner of the blob bounding box is denoted as \((a, b)\), the center of position parameters \((x^{(n)}, y^{(n)})\) is at \((a+c/2, b+d/2)\), with \(c\) and \(d\) the width and length of the blob bounding box along the \(x\) and \(y\) directions. It makes sure that the distribution of samples is centered at the upper middle of the blob bounding box (Figure 2.3 (b)).

Particle filtering is an iterative process, which can also be extended to successive images in a video sequence for object tracking [63]. Since there exists a large correlation between consecutive video frames, detection results from the previous frame, such as position and scale of the person, are very relevant to that of the current frame. At the same time the current frame may differ from the previous frame, so a drift term is introduced to account for the new information in the current frame.
2.3. Methodology

Although several feature points such as head top, head center, torso center, torso bottom and both shoulders can be estimated from the 2D model, our pose recognition system only use the torso center to present the person’s location. The extraction of additional features is outlined in the next section.

2.3.3 Hand detection and tracking

In addition to the 2D model mentioned in Section 2.3.2 for human’s torso detection and tracking, foreground pixels are further segmented into skin-color and non-skin-color regions. A skin color model in the RGB color-space is used to select skin color pixels on the foreground image. This human skin color model is similar to the model in [19]. If foreground pixels mapping into the RGB color-space satisfy the following conditions, they will be considered as skin-color pixels.

\[
\left| \arctan \left( \frac{B}{R} \right) - \frac{\pi}{4} \right| < \frac{\pi}{8}, \quad \left| \arctan \left( \frac{G}{R} \right) - \frac{\pi}{18} \right| < \frac{\pi}{18}, \quad \left| \arctan \left( \frac{B}{G} \right) - \frac{\pi}{5} \right| < \frac{\pi}{15}
\] (2.1)

After the skin color pixels are selected, two post-processing steps are used to get rid of false positive detections. The first step is to delete regions with very small size, which are impossible to be face and hand regions. In the second step, a motion mask is introduced to exclude regions which are far way from previous hand locations. It limits the movement of hands within a certain bounding box. Additionally, the face region can be separated from the hands regions, either by using the size of the connected skin color area, or the head location estimated from Section 2.3.2. In our approach, the size information is used, which is enough to distinguish the face region from the hand regions. From the remaining blobs, we calculate the centers of gravity and use them to represent hand positions.
2.3.4 3D reconstruction

Until now the obtained torso center and both hand positions are from a single view, but the method can be easily applied to other views as well. The multiple camera setups are shown in Figure 2.4; there are three cameras in total. One of the cameras (camera 2 in Figure 2.4) is located at the front of the recording room, which captures the frontal view of the user. The other two (camera 1 and camera 3 in Figure 2.4) are in the corners of the room. They give two side views of the user.

Since these three cameras are synchronized, the 3D positions of a torso and hands of a human body can be obtained by using calibrated cameras. As for the hand position, we make the assumption that the left hand is always on the left side of the torso for all three views, and the right hand is on the right side of the torso. Therefore, left hand and right hand positions can be identified for all views and used to calculate 3D positions.

2.3.5 Feature space construction

The input of the proposed pose recognition system are 2D (frontal view camera) and 3D positions of the torso center and the hands. However, we transfer them into normalized feature space and train the classifier in this new feature space. The reason is that the pose recognition system should be scene invariant. That is, no matter where the person is in the scene, or how far the person is from the cameras, the predefined key poses should be recognized. Therefore the feature space is built by using relative positions between hands and torso center, such as distances and angles. Based on this, we construct 20 feature components as follows, which are denoted as \( F_{\text{set}} = \{c_1, c_2, \ldots, c_{20}\} \):

\[
\begin{align*}
c_1 &= \frac{(x'_2 - x'_3)}{s}, & c_2 &= \frac{(y'_2 - y'_3)}{s}, & c_3 &= \frac{(x'_2 - x'_3)}{s}, & c_4 &= \frac{(y'_2 - y'_3)}{s}, \\
c_5 &= \arctan \frac{y'_2 - y'_3}{x'_2 - x'_3}, & c_6 &= \arctan \frac{y'_2 - y'_3}{x'_2 - x'_3},
\end{align*}
\]
2.4 Experiments

\[ c_7 = x_3^l - x_3^r, \quad c_8 = y_3^l - y_3^r, \quad c_9 = z_3^l - z_3^r, \]
\[ c_{10} = x_3^l - x_3^l, \quad c_{11} = y_3^l - y_3^l, \quad c_{12} = z_3^l - z_3^l, \]
\[ c_{13} = \frac{x_3^l - x_3^l}{s}, \quad c_{14} = \frac{y_3^l - y_3^l}{s}, \quad c_{15} = \frac{z_3^l - z_3^l}{s}, \]
\[ c_{16} = \frac{x_3^r - x_3^r}{s}, \quad c_{17} = \frac{y_3^r - y_3^r}{s}, \quad c_{18} = \frac{z_3^r - z_3^r}{s}, \]
\[ c_{19} = \sqrt{(x_3^l - x_3^l)^2 + (y_3^l - y_3^l)^2 + (z_3^l - z_3^l)^2}, \quad c_{20} = \sqrt{(x_3^r - x_3^r)^2 + (y_3^r - y_3^r)^2 + (z_3^r - z_3^r)^2} / s. \]

Here \((x_3^l, y_3^l, z_3^l)\), \((x_3^l, y_3^l, z_3^l)\), \((x_3^r, y_3^r, z_3^r)\) and \((x_3^r, y_3^r, z_3^r)\) are the 2D positions of the torso center, left hand and right hand. \((x_3^l, y_3^l, z_3^l)\), \((x_3^l, y_3^l, z_3^l)\), \((x_3^r, y_3^r, z_3^r)\) are the 3D positions of the torso center, left hand and right hand. The obtained scale parameter in the 2D model \(P = (x, y, scale)\) is indicated by \(s\). The classifier will be trained and tested on this defined feature space \(F_{set}\).

2.4 Experiments

2.4.1 Video recording

Before the start of the video recording, we take some snapshots for the purpose of calibration. We recorded videos of 15 volunteers from 6 races (Netherlands, China, France, Italy, Turkey, and Syria). Five of them are female and the others are male. The predefined key poses are shown in Figure 2.5. The images are from the frontal view camera (camera 2 in Figure 2.4).

2.4.2 Implementation

In the background subtraction implementation, the number of the Gaussian models is chosen to be 3. In order to exclude shadows from the foreground image, a shadow removing approach is also employed. As for the particle filter, we choose the number of particles to be 500, which is a trade-off between precision and computing time.

2.4.3 Pose classification

The key poses are designed for gaming control, so they should be easy for users to remember and perform. The number of the poses should not be too high; otherwise it also increases the difficulty for users. In our system, we defined nine poses in total, as shown in Figure 2.5. From top row to bottom row and left to right, these nine poses are labeled as pose1 to pose9. In order to build a classifier, we manually labeled the frames containing the nine poses into nine classes. For each pose, the samples are selected from each of the 15 persons. Our experimental data set contains 1515 samples of 9 pose types (classes) and 20 features. On average, each pose class is represented by 170 samples.
2.5 Results and discussions

We evaluated pose classifiers using two cross-validation approaches. The first one is the leave-one-person-out (LOPO) where in each step (fold) we leave out all the samples corresponding to one person as the test set and use the samples of the remaining 14 persons for training the pose classifier. The LOPO procedure is repeated 15 times (folds). The other approach is randomly splitting each of the nine classes into 15 parts, using 14 as training set and one as testing set. We call the second approach 15-fold rotation (FORO). We compared performances of several statistical classifiers with different complexities. Specifically, we evaluated the nearest mean classifier (NMC), the linear classifier (LDC) and the quadratic classifier (QDC) assuming normal densities and the non-parametric Parzen classifier. The QDC and Parzen were built either directly in the 20D feature space or in the 8D subspace derived by a supervised feature extractor: linear discriminant analysis (LDA). The results are shown in Table 2.1.

As shown in Table 2.1, there is a clear separation between pre-defined poses. The differences between LOPO and FORO illustrate that the latter is highly optimistically biased. The reason is that similar examples extracted from neighboring frames of one person may
Table 2.1: Cross-validation results of pose classifiers (mean errors with standard deviation).

<table>
<thead>
<tr>
<th>method</th>
<th>LOPO</th>
<th>FORO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean pose err.</td>
<td>max pose err.</td>
</tr>
<tr>
<td>NMC</td>
<td>0.06(0.09)</td>
<td>0.18(0.35)</td>
</tr>
<tr>
<td>LDC</td>
<td>0.06(0.07)</td>
<td>0.14(0.35)</td>
</tr>
<tr>
<td>QDC</td>
<td>0.10(0.11)</td>
<td>0.23(0.34)</td>
</tr>
<tr>
<td>LDA+QDC</td>
<td>0.07(0.09)</td>
<td>0.16(0.35)</td>
</tr>
<tr>
<td>Parzen</td>
<td>0.07(0.09)</td>
<td>0.16(0.35)</td>
</tr>
<tr>
<td>LDA+Parzen</td>
<td>0.06(0.07)</td>
<td>0.14(0.35)</td>
</tr>
</tbody>
</table>

Table 2.2: Confusion matrices of nine poses.

<table>
<thead>
<tr>
<th>True Labels</th>
<th>Estimated Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P1</td>
</tr>
<tr>
<td>P1</td>
<td>198</td>
</tr>
<tr>
<td>P2</td>
<td>0</td>
</tr>
<tr>
<td>P3</td>
<td>2</td>
</tr>
<tr>
<td>P4</td>
<td>0</td>
</tr>
<tr>
<td>P5</td>
<td>1</td>
</tr>
<tr>
<td>P6</td>
<td>2</td>
</tr>
<tr>
<td>P7</td>
<td>0</td>
</tr>
<tr>
<td>P8</td>
<td>0</td>
</tr>
<tr>
<td>P9</td>
<td>0</td>
</tr>
</tbody>
</table>

end up in both training and test set. It is also interesting to notice that this difference grows with classifier complexity, which is a clear sign of over-fitting. We observe that the simplest method (NMC) provides comparable performance to more complex classifiers which need an extra dimensionality reduction step to avoid the curse of dimensionality. We conclude that the extracted features are informative and do not require use of more complex classifiers.

We also calculate the confusion matrices of the 9-class pose classifier (NMC). The results are shown in Table 2.2, which is the sum of 15 per-fold (person) LOPO confusion matrices. As can be seen from Table 2.2, the results are promising. Most of the poses can be recognized very well. However there is quite a large error between pose 4 and pose 6 and all the 20 misclassified samples are from the same person. We searched back in the dataset and found that the 3D positions of the person’s right hand in the 20 samples are not correct due to wrong detections. We conclude that the wrong representation/detection of the feature points is the reason for the misclassification.
2.6 Spatial game application

2.6.1 Implementation

As an application for the pose recognition system we implemented a spatial game, based on the proposal of Phong in [82]. This is a variation of the game Pong [83] in which the player controls a bat to bounce off balls. In Phong the player controls a chameleon which has to bounce off photons, see in Figure 2.6. The position of the chameleon is determined by the player’s position in front of the camera. The photons can have 6 different colors: red, blue, green, yellow, cyan and magenta. The chameleon can change into each of these colors when the player adopts the appropriate pose.

When the photon hits the ceiling, it changes color. When the photon is bounced off while the chameleon has the wrong color, the controls flip. Left becomes right and vice versa. When the chameleon has the right color while bouncing the photon off, the score and speed of the photon are increased. When the chameleon misses the photon the ground is heated up. After 4 misses the ground is too hot and the game is over. At random moments a bug flies into the scene which can be eaten by the chameleon when the player adapts to the eating pose. This will increase the score and the ground will cool down.

The game is implemented using the graphics engine Ogre [84]. The input for the game is given by the pose recognition system. The pose recognition system and the spatial game are two separate applications which communicate via sockets [85]. Therefore, it is possible for the two applications to run on different computers and communicate through a network. The
2.6. Spatial game application

Figure 2.7: Spatial game interface. On the left side is the interface of the game, which shows the level, bounces, heat and score of the player. The three windows on the right side are the results from vision-based analysis. From top to bottom, they are original image, results from body parts segmentation and pose recognition, and foreground binary image.

pose recognition system sends two types of data to the spatial game in every time step. It sends an integer that represents a pose (1-9) and an integer representing the 1D-location of the player. Whenever the spatial game receives this data, it updates the position of the chameleon according to the position integer and it carries out the action belonging to the pose index that was send. These actions consist of 6 poses for changing the chameleon into the 6 different colors, 1 pose is for eating the bug, 1 pose is for starting the game and 1 pose is to pause the game. Figure 2.7 gives a screen shot of user playing the game. On the left side is the interface of the game, which shows the level, bounces, heat and score of the player. The three windows on the right side are the results from vision-based analysis. From top to bottom, they are original image, results from body parts segmentation and pose recognition, and foreground binary image.

2.6.2 Discussions

In our first test runs it became clear that the sensitivity of the pose recognition to detect the change of poses gave a problem for the game player. Whenever the player needs to change from one pose to another there could be a different pose adopted that is “in between” these
two poses. When this happens the color of the chameleon in the game is shortly changed into an unwanted color. This problem has been overcome by using a counter whenever a new pose is adopted. The new pose has to be adopted for 4 consecutive time steps until its corresponding action is carried out. This adjustment improved the playability of the game as the user feels having a better control of the chameleon. We did encounter a short delay in handling the players input. The delay is caused by the image processing time. This is mostly noticed with updating the chameleon’s position by the player’s actual location, but the delay is too small to actually cause gameplay problems. The implementation of the Phong game showed that the gameplay of the spatial game is interesting. As a next step it is good to reduce the delay to a minimum. After this improvement it is interesting to create a more complex game with an elaborate user interface.

2.7 Conclusions

In this paper, we present an approach to capture markerless human motions and recognize human poses. By transferring the 2D and 3D positions of the selected feature points into a normalized feature space, a simple classifier is shown to be sufficient for multi-pose recognition. This is also quite attractive from a computational point of view. The processing time of each frame is 0.047 seconds, including background subtraction, torso and hand detection, and pose recognition. However, due to the small number of the selected feature points, some errors are introduced in the pose classification. Therefore, in our future work, we will focus on extracting more relevant features to improve the performance of the classifier. Moreover, we will investigate detectors to reject non-pose examples based on the proposed features.

We also described a real-time computer vision based application: a spatial game system. This pose-driven spatial game is a real time man-machine interaction without obtrusive sensors. It shows the possibility of a new way of interactions in novel computer games and entertainment. The combination of computer vision research and a practical application is quite useful. It allows us to directly test if the proposed algorithm satisfies certain requirements, in a specific application environment. Future work will include improving the robustness of the system (e.g. better skin color detection, more robust feature detection) and developing multiple-user applications. One of the challenges will be to solve the occlusion problem if users are allowed to move freely.

Acknowledgements

This research has been supported by the GATE (Game Research for Training and Entertainment) project, funded by the Netherlands Organization for Scientific Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie). We used PRTools toolbox (http://prtools.org) and PRSD Studio (http://prsdstudio.com) software packages to build classifiers.
Chapter 3

Real Time Multiple People Tracking and Pose Estimation

Abstract

In this paper we present a combined probability estimation approach to detect and track multiple people for pose estimation at the same time. It can deal with partial and total occlusion between persons by adding torso appearance to the tracker. Moreover, the upper body of each individual is further segmented into head, torso, upper arm and lower arm in a hierarchical way. The simplicity of the feature and the simplified model allow close real time performance of the tracker. The experimental results show that the proposed method can deal with most of the inter-person occlusions, as well as certain self occlusions. It's also much faster than the existing methods with comparable accuracy.

This chapter has been published as:
Chapter 3. Real Time Multiple People Tracking and Pose Estimation

3.1 Introduction

Currently, multiple people tracking and pose estimation has drawn more and more attention due to its large applications in surveillance, pose recognition, understanding interactions between persons [16, 93-95]. Compared with a single person situation [38, 60, 91, 92], multiple person tracking and pose estimation has more challenges, such as dealing with inter-person occlusions (occlusions between persons) and self occlusions. The occlusion is always a difficult problem to deal with, due to its hard to predict behavior. For instance, persons may change their directions after occlusion or still move in the same direction. Apart from this, the appearance of people may also change, such as from frontal view to lateral view. Additionally when people are very close to each other, it is quite easy to cause shadows on the body. All of this makes it a challenge to cope with people occlusion in a tracking system.

3.2 Related work

There is a large amount of research on multiple people tracking and pose estimation. In [86], they propose an approach for automatic initialization and tracking of human poses. It combines bottom-up evidence with top-down priors to get efficient pose estimation. The proposed algorithm can handle most of the self occlusions by using the appearance model and the occlusion map. However, the runtime is around 45 seconds per frame due to the search of possible part configurations. It is not very suitable for real time applications. In [39], the appearance model of a person is learned from the video data. Then the person is tracked by detecting the learned model in each frame. The advantage of the approach is that people can be accurately tracked from a single view in front of complex backgrounds. But the appearance model needs to be built before the online tracking.

In this paper, we introduce a fast multiple people tracking and pose estimation approach by combing several probabilities derived from image observation. The flowchart of the proposed system is shown in Figure 3.1. In the first step, the system detects different body parts by using an initial pose. The appearance model of torso and arm is build. In the second step, different body parts are segmented in a hierarchical way for body part detection and tracking. In the last step, in order to deal with inter-person occlusions, torso color histogram is used to distinguish different persons. A combined probability approach is used to estimate the upper-body pose. The main contributions of our paper are, tracking of multiple persons, dealing with inter-person occlusions and body parts segmentation in a real time system.

3.3 Methodology

In our approach, a generative body template is used to represent the upper body configuration as shown in Figure 3.2. It is composed of two parts, a 2D-upper-body model of torso and head [81], and four 2D rectangles for upper and lower arms. The parameters in the 2D upper-body-model are torso position and scale and the model is described as \( h = \{ x_t, y_t, \text{scale} \} \).
The upper and lower arms are modeled by image patches $a_u = \{x_u, y_u, \theta_1\}$ and $a_i = \{x_i, y_i, \theta_2\}$. Here $(x_t, y_t)$, $(x_s, y_s)$ and $(x_e, y_e)$ are the $x$ and $y$ coordinates of torso center, shoulder and elbow. $\theta_1$ and $\theta_2$ are the angles of upper and lower arms with respect to the main torso direction. Assuming there is a single person in the current frame $t$, the parameters for different body parts are put together into one state vector:

$$X_t = \{h, a_u^l, a_i^l, a_u^r, a_i^r\}$$

where $h$ represents torso and head parameters, $a_u^l$ represents left upper arm parameters, $a_i^l$ represents left lower arm parameters, $a_u^r$ represents right upper arm parameters, $a_i^r$ represents right lower arm parameters. We define the prior probability of upper body pose as:

$$p(X_t) = p(h, a_u^l, a_i^l, a_u^r, a_i^r)$$

$$= p(h) \cdot p(a_u^l, a_i^l, a_u^r, a_i^r | h)$$

$$= p(h) \cdot p(a_u^l, a_i^l | h) \cdot p(a_u^r, a_i^r | h)$$

$$= p(X_{t,1}) \cdot p(X_{t,2}) \cdot p(X_{t,3})$$
The prior probability \( p(h) \) is assumed to be Gaussian. The candidate sample set is generated on the foreground blob with different positions and scales. The prior probability of left arm conditioned on torso \( p(a'_l, a'_l | h) \) is derived from the loose connection between the start point of left upper arm and left shoulder position. Here we use a Gaussian distribution to model \( p(a'_l, a'_l | h) \). It also can be other forms of distributions, such as uniform distribution, or more complex distribution learned from training data [37]. The prior probability of right arm conditioned on torso \( p(a'_r, a'_r | h) \) is also a Gaussian. Although a similar upper body model has been used more often for both 2D and 3D pose estimation [37, 41, 43, 89, 90], the main difference of our approach is that the model is described in a hierarchical way. Instead of putting all the parameters into one state vector \( X_t \), they are split into three state vector vectors:

\[
X_{t,1} = \{h\} \\
X_{t,2} = \{a'_l, a'_l\} \\
X_{t,3} = \{a'_r, a'_r\}
\]  

(3.3)

In this model, there are two assumptions: one is that both state vector \( X_{t,2} \) and \( X_{t,3} \) are depend on state vector \( X_{t,1} \); the other is that state vector \( X_{t,2} \) and \( X_{t,3} \) are independent. The first assumption is motivated by the fact that left arm and right arm should always be connected to the torso through shoulder joints. The second assumption is based on kinematic constrains. We assume that the movement of the person’s left arm is not related to the movement of a person’s right arm if the person is allowed to move freely. In certain cases, such as waking, running, jumping, etc, there are correlations between the movements of left arm and right arm. However, we do not want to limit the model to a certain motion.

A strong motivation of our model is that it can describe the desired probability as accurate as possible, but at the same time allow for real time processing. The proposed approach simplifies the model and reduces the state vector dimension. When particle filtering is used to estimate the state of a system, this advantage becomes more obvious, because the computation time is directly related to the dimension of the state. Since state vector \( X_{t,2} \) and \( X_{t,3} \) are dependent on state vector \( X_{t,1} \), a hierarchical way can be used to first segment torso and head, then left and right arms. The prior distributions of these three state vectors are
indicated as \( p(X_{t,1}) \), \( p(X_{t,2}) \), \( p(X_{t,3}) \). The image likelihood function of the candidate state vectors will be discussed in Section 3.3.3.

### 3.3.1 Initialization step

In the initialization, the system automatically detects different body parts by using an initial pose. The initial pose is shown in Figure 3.3. From the prior knowledge of this initial pose and from general geometrical properties of human beings, shoulder, hand and elbow positions are extracted in this initial frame, to be used for segmenting different body parts.

When people are appearing in the scene, the 2D-upper-body model is used to fit the person’s head and torso on a foreground binary image. The foreground image is obtained by using background subtraction. The background image is built with a mixture of \( k \) Gaussian models [80]. In order to exclude shadows from the foreground image, a shadow removing approach is also employed [19]. From the 2D model, we can roughly locate the person’s shoulders \((x_s, y_s)\). Then a simple skin color model is utilized to detect person’s hands \((x_h, y_h)\) [19]. The initial pose is defined as people stretching both of their arms sideway. When the distance between person’s two hands is larger than a predefined threshold, the person is assumed performing the initial pose. In this case, the full arm length (FAL) can be obtained, which is the Euclidian distance between the shoulder \((x_s, y_s)\) and the hand \((x_h, y_h)\) on the same side of the body. The full arm length is equal to:

\[
FAL = \sqrt{(x_s - x_h)^2 + (y_s - y_h)^2}
\]  

(3.4)

Since shoulder and hand are the two end points of one arm, the elbow can be considered as the middle point of the arm, which is \((x_e, y_e) = \text{mean}[(x_s, y_s), (x_h, y_h)]\). These body joints, shoulder, elbow and hand, are used to segment torso, upper arm and lower arm from the whole body configuration.
3.3.2 Appearance model building

From the previous initialization, torso, upper arm and lower arm are segmented. In order to deal with inter-person occlusions and self occlusions, an appearance model for torso and arm is built. In our approach, a color histogram in HSV color space is used for modeling torso appearance. For the arm appearance modeling, a color vector in normalized RGB color space is used. It is a trade-off between accuracy and efficiency.

3.3.2.1 Torso appearance model

For torso modeling, the original image is first transferred from RGB to HSV color space. Then \( H, S \) components are used to construct a 2D color histogram in order to exclude the influence of lighting conditions. The color histogram of the torso model is represented as \( H_t \). We use the Bhattacharyya distance \([87]\) to measure the appearance similarity between the torso appearance model \( H_t \) and candidate models \( H_c \). It is defined as:

\[
d(H_t, H_c) = \sqrt{1 - \sum_{i=1}^{m} \sqrt{H_t(i) \cdot H_c(i)}}
\]  

where \( H(i) \) is the normalized histogram value of the \( i^{th} \) bin. \( m \) is the total number of bins in the histogram. The Bhattacharyya distance is a number between 0 and 1. The value of 0 means a perfect match, while 1 is a total mismatch. The Bhattacharyya distance is used to derive the probability of the torso model appearing in the scene.

3.3.2.2 Arm appearance model

For arm modeling, the color vector is composed of normalized \( r \) and \( g \) component in RGB color space \([86]\). The color vector of the target arm model is described as \( V_t \):

\[
V_t = (r_1', g_1', r_2', g_2', ..., r_m', g_m')
\]  

\( r_i' \), \( g_i' \) are the mean \( r \) and \( g \) component of the \( i^{th} \) part in an image patch. \( m \) is the total number of partitions along the middle line of the candidate image patch \( \alpha_u = \{x_u, y_u, \theta_u\} \) and \( \alpha_i = \{x_i, y_i, \theta_i\} \), as shown in Figure 3.4. Here an image patch is evenly segmented into \( m \) parts along the middle line. The value of \( m \) depends on the resolution of the image and the size of the image patch.

Since the size of candidate image patches for upper arms and lower arms is much smaller than the torso, using color vector representation gives comparable accuracy as color histograms. However, calculating a color vector is faster than a color histogram. Specifically, when the number of candidate image patches is large, this advantage becomes important for real time application.
3.3 Methodology

3.3.3 Body parts segmentation

After initialization and appearance modeling, a hierarchical method is used to segment different body parts in the subsequent video frames. For each frame, first head and torso region are segmented by using the 2D upper-body model, since head and torso give more robust visual cues, compared with other relative small body parts. From this 2D model, we can roughly locate the head, torso and shoulder position. The position of shoulder serves as the start point of the upper arm. In order to locate the upper arm, image patches \( a_u = \{x_s, y_s, \theta_s\} \) are generated around the shoulder position \((x_s, y_s)\). The length of these image patches (LIP) is equal to the half length of the full arm, denoted as \( LIP = \frac{FAL}{2} \). There is no large variation of the length, since most poses are in a 2D plane [88]. The width of the image patch (WIP) is kept constant. After the upper arm is located, the end point of the upper arm is supposed to be an elbow. Then the elbow position \((x_e, y_e)\) will be used as the starting point for lower arm searching. Image patches \( a_l = \{x_e, y_e, \theta_e\} \) are generated in the same way as for the upper arm.

The body part segmentation and tracking is formulated as the problem of state estimation. A combined probability estimation approach is used to track and segment each individual. For each person in one frame \( I_t \), three individual particle filters are used to estimate 2D poses in a hierarchical way. State vector \( X_{t,1} \) is estimated first. The image likelihood function \( p(I_t | X'_{t,1}) \) of a state candidate \( X'_{t,1} \) is estimated as:

\[
p(I_t | X'_{t,1}) \propto \exp\left(-\left(1 - \rho_t\right) - \left(1 - \rho_d\right) \right)
\]

where \( \rho_t \) is the fitness coefficient of the 2D upper-body model on the foreground binary image [81]. Therefore, the first term gives the probability of how the torso and head model fits on the foreground image. \( \rho_d \) is related to the torso appearance of the person. As mentioned in Section 3.3.2, the torso appearance is modeled by a 2D color histogram \( H_t \) in HSV color space and is measured by Bhattacharyya distance. \( \rho_d \) is given by:
\[ \rho_j = 1 - d(H_j, H_c) = 1 - \sqrt{1 - \sum_{i=1}^{m} H(i) \cdot H_c(i)} \]  

(3.8)

After estimating state vector \( X_{t,1} \), state candidates \( X'_{t,2} \) and \( X'_{t,3} \) are generated. They are evaluated by image likelihood functions \( p(I_t | X'_{t,2}) \) and \( p(I_t | X'_{t,3}) \) respectively. \( p(I_t | X'_{t,2}) \) is given by:

\[ p(I_t | X'_{t,2}) \propto \exp \left( -\alpha \cdot (1 - \rho_f) - \beta \cdot (1 - \rho_c) - \gamma \cdot (1 - \rho_e) \right) \]  

(3.9)

where \( \rho_f \) indicates how probably image patches fit on the foreground. It is described as:

\[ \rho_f = \frac{NFP[a = \{ x, y, \theta \}]}{LIP \times WIP} \]  

(3.10)

\( NFP \) counts the number of foreground pixels within image patch \( a = \{ x, y, \theta \} \). \( \rho_c \) gives the appearance similarity between the target color vector \( V_t \) and the candidate image patch color vector \( V_c \):

\[ \rho_c = 1 - \frac{1}{m} \sum_{i=1}^{m} \sqrt{\left( r_i' - r_i^c \right)^2 + \left( g_i' - g_i^c \right)^2} \]  

(3.11)

\( \rho_e \) is related to image patch edges. It is defined as the response to a parallel line filter, as in [86]. Instead of adding these three terms directly, we give each of them a weight \( \alpha \), \( \beta \) and \( \gamma \). The value of the weight is not constant, but depends on the scene. For instance, if occlusion is detected, \( \alpha \) is set to a small value, since \( \rho_f \) will become unreliable. In the contrast, the value of \( \beta \) and \( \gamma \) will increase. So the weighting of different probabilities is related to the occlusion estimation and improves the robustness and accuracy of arm segmentation.

The calculation of \( p(I_t | X'_{t,3}) \) is the same as \( p(I_t | X'_{t,2}) \). After all the state candidates \( X'_{t,1} \), \( X'_{t,2} \) and \( X'_{t,3} \) are evaluated, the maximum a posterior poses at frame \( t \) can be computed from:

\[ \{ \hat{X}_{t,1}^{MAP} \} = \max_{X_{t,1}} \left( p(I_t | X'_{t,1}) \right) \]
\[ \{ \hat{X}_{t,2}^{MAP} \} = \max_{X_{t,2}} \left( p(I_t | X'_{t,2}) \right) \]
\[ \{ \hat{X}_{t,3}^{MAP} \} = \max_{X_{t,3}} \left( p(I_t | X'_{t,3}) \right) \]  

(3.12)
3.3. Methodology

The posterior probability distribution $p(X'_{t,1} | I_t)$ for the state vector $X'_{t,1}$, at the current frame $t$, together with the temporal diffusion model $p(X'_{t+1,1} | X'_{t,1})$ forms the prior probability distribution $p(X'_{t+1,1})$ for the next frame $t+1$. Since the movement of a person is not very predictable, $p(X'_{t+1,1} | X'_{t,1})$ is simplified as a Gaussian model. The posterior probability distributions $p(X'_{t,2} | I_t)$, $p(X'_{t,3} | I_t)$ are calculated in the same way.

$$p(X'_{t+1,1}) \propto p(X'_{t,1} | I_t) p(X'_{t+1,1} | X'_{t,1}) \quad (3.13)$$

$$p(X'_{t,1} | I_t) \propto p(I_t | X'_{t,1}) p(X'_{t,1}) \quad (3.14)$$

3.3.4 Multiple people tracking

When there are multiple persons in the scene, we can simply track them separately. For the remaining of the paper, we assume to deal with a two persons’ scenario. However, this method can be easily extended to a situation with more than two persons. As soon as two persons are appearing in the scene (assume they do not have any occlusion in the initialization step), the appearance model of their torso $H_t$ is built (in Section 3.3.2). When there is an overlap between these two persons, the similarity of torso color histogram is calculated for both of them. It will indicate which person is in the front and which person is occluded. The front person is always tracked. The body part segmentation of the front person is the same as described in Section 3.3.3. An example frame of the front person being tracked is given in Figure 3.5. The back (occluded) person is only tracked when he/she is appearing again. During occlusion, the joint locations of the back person maintain the same as in the last frame before occlusion.
3.4 Experimental results

In order to test the proposed approach, we recorded two video sequences in our own lab and used one from the Multimedia and Geometry group in Utrecht University. They all have cluttered background. The video starts with empty background, and then people walk into the scene. When people are standing in the middle of the scene and performing the initial pose (Figure 3.3), the system automatically starts pose tracking. People are facing the camera and allowed to have large movement of their arm and whole body. The recorded frame rate is 25 frames per second. We evaluated our approach against ground truth. The ground truth is obtained by manually labeling the body joints: head center, torso center, left elbow, left wrist, right elbow, and right wrist. The joint error (in pixel units) is used to quantitatively measure the segmentation accuracy and is defined as follows:

\[
\text{Error} = \sqrt{(x_j - x_g)^2 + (y_j - y_g)^2}
\]

where \((x_j, y_j)\) is the obtained joint location, \((x_g, y_g)\) is the manually labeled ground truth. For arm segmentation, we compared the method with occlusion estimation (line with circles in Figure 3.6) to that without occlusion estimation (line with dots in Figure 3.6).

For the first approach, the value of \(\alpha\), \(\beta\), and \(\gamma\) in \(p(I_t | X'_{r,s})\) will be changed when the occlusion is detected. For the second approach, the value of \(\alpha\), \(\beta\), and \(\gamma\) are kept constant and no occlusion is taken into account. The values of \(\alpha\), \(\beta\), and \(\gamma\) are chosen as 0.5, 0.2, and 0.3. When occlusion is detected, they are modified to 0.1, 0.7, and 0.2 in the first approach. These values were empirically found to work well. The average joint error is used to measure the segmentation accuracy.

The results are shown in Figure 3.6. Figure 3.6 (a) is the segmentation results of person1 (front person), while Figure 3.6 (b) is the results of person2 (back or occluded person). From the results we can see that adapting the weight values of person1 improves the pose estimation in case of occlusion. For person2, since he is occluded, the joint locations just copied the ones in the last frame before occlusion. Therefore, the errors are the same for the two approaches, as shown in Figure 3.6 (b).

In Figure 3.7, some images from our test data set are shown. The four frames in the left column show the segmentation results of using occlusion estimation. The right column shows the results without using occlusion estimation. From these frames, we can see that although the persons are wearing clothes with texture or similar color, the proposed approach can handle it quite well. The processing speed of our current implementation is 10 to 13 frames per second using a 2.66GHz Core Duo PC. With optimized implementation, the speed still can be increased. Here we only showed the experimental results from one video sequence. Similar results were obtained with the other videos.
3.4. Experimental results

Figure 3.6: Comparison of performance dealing with occlusion. The error is measured in pixels. (a) The segmentation results of person1. (b) The segmentation results of person2.
Figure 3.7: Pose estimation results with (left column) and without (right column) occlusion estimation.
3.5 Conclusions

In this paper, we introduced a combined probability approach to estimate human upper-body configuration in real time. It can deal with inter-person occlusions and certain self occlusions using image features, such as foreground, edge and appearance. Instead of weighting these image features equally, we give each of the features a weight according to the occlusion estimation. For instance, when inter-person occlusion is detected, we increase the weight of the appearance and the edge features and reduce the weight of the foreground feature. The reason is that when there is occlusion between two persons, their foreground silhouette also has overlap. The foreground feature becomes unreliable to distinguish the two persons. We showed that using occlusion adaptive probability weights for arm segmentation improved the upper and lower arm position estimation. The hierarchical way of upper body segmentation reduces the dimensionality of state vector. When particle filtering is used to estimate the state, this advantage becomes obvious, because the computational time is directly related to the dimensionality of the state. With the hierarchical approach, multiple people pose tracking can be realized in real time, which is suitable for practical applications. We will further extend multiple people pose estimation into pose recognition.

Acknowledgements

This research has been supported by the GATE (Game Research for Training and Entertainment) project, funded by the Netherlands Organization for Scientific Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie).
Chapter 4

Multiple People Tracking and Pose Estimation
with Occlusion Estimation

Abstract

Simultaneously tracking poses of multiple people is a difficult problem because of inter-
person occlusions and self occlusions. This paper presents an approach that circumvents this
problem by performing tracking based on observations from multiple wide-baseline cameras.
The proposed global occlusion estimation approach can deal with severe inter-person
occlusions in one or more views by exploiting information from other views. Image features
from non-occluded views are given more weight than image features from occluded views.
Self occlusion is handled by local occlusion estimation. The local occlusion estimation is
used to update the image likelihood function by sorting body parts as a function of distance
to the cameras. The combination of the global and the local occlusion estimation leads to
accurate tracking results at much lower computational costs. We evaluate the performance of
our approach on a pose estimation data set in which inter-person and self occlusions are
present. The results of our experiments show that our approach is able to robustly track
multiple people during large movement with severe inter-person occlusions and self
occlusions, whilst maintaining near real-time performance.

This chapter has been published as:
F. Huo and E.A. Hendriks. Multiple people tracking and pose estimation with occlusion
4.1 Introduction

Video-based applications such as visual surveillance, human computer interaction (HCI) and serious games have become more and more wide-spread in our daily lives [16, 17, 95, 96]. One of the key components of realizing these applications is person detection, tracking and pose estimation [78, 99, 100]. Multiple people tracking and pose estimation is one of the most challenging topics compared to single person tracking and pose estimation, due to occlusions between persons and self occlusions. The occlusion is a difficult problem to deal with, since it is hard to predict people’s behavior. For instance, people may change their moving directions after occlusion or still move in the same direction. Apart from this, the appearance of people may also change, such as from frontal view to lateral view. Moreover, when people are very close to each other, it is quite easy to cause shadows on each other’s body. All of this makes it a challenge to cope with people occlusion in a tracking system. Many of the existing methods are aiming to deal with self occlusions, or with partial occlusions between individuals [32, 35]. They are not suitable to deal with severe occlusions between individuals. By severe occlusions, we mean that more than 50% of the body part is occluded. However, the severe occlusion is a very common scenario in indoor or outdoor environments [46, 97]. It is challenging and interesting to steer research in this direction. Another aspect is that many existing tracking systems are only suitable for offline applications, due to their computational complexity. Therefore, reducing processing time or decreasing search space dimensionality of tracking systems is required for certain applications. Examples of such applications are pose-driven spatial games, in which players get rid of controllers and play games using intuitive body movements and poses [60], and serious games in which people have to interact with a simulated virtual world in real-time.

In this paper, we present a multiple view approach to track multiple people simultaneously and estimate their poses in 3D. Our goal is to robustly track individuals in a small group of people and realize fast 3D pose estimation. The tracked poses are not limited to certain actions, such as walking, running, or jumping. People freely move their body and limbs and as a result, they perform a large variety of poses. Our emphasis is on tracking the upper body and arms, since in general upper body poses are most frequently used in HCI applications. The tested video sequences are recorded indoors, but without specific constraints onto lighting or background. The three main contributions of our work are as follows:

1. The proposed approach can simultaneously track poses of multiple people with severe occlusions in real-time.

Unlike most of the previous approaches, our method not only can deal with partial occlusions between persons, but also with severe occlusions in an efficient way. The proposed approach utilizes reliable information from non-occluded views to reduce ambiguity in occluded views. In this way, it improves tracking accuracy by combining information from multiple views. The 3D human pose that fits best on 2D image features is selected as the final estimated pose. The proposed approach can track up to three persons simultaneously.

2. An integrated global and local occlusion estimation is used for handling inter-person occlusions and self occlusions.
The global and local occlusion estimation is combined in such a way that it is suitable for real-time performance. The global occlusion estimation is a person-level occlusion estimation, which handles severe occlusions between persons. The local occlusion estimation is used for dealing with occlusions by different body parts of a person. The combination of the global occlusion estimation and the local occlusion estimation helps us to obtain accurate tracking results.

3. A part-based hierarchical model combined with particle filtering is used which drastically reduces computation complexity.

Instead of putting all the separate body parts, such as head, torso and arms, into one framework and estimating all the parameters at the same time, we split the framework into two stages. In the first stage, a person’s head and torso is tracked and located. In the second stage, we search for the person’s arms based on the expected shoulder locations. In this way, the dimensionality of the search space for body configuration is decreased. Particle filters are used to estimate posterior distributions of different body part configurations.

The paper is organized as follows. Section 4.2 gives an overview of related work. We discuss the part-based upper body model in Section 4.3 and discuss the hierarchical tracking and pose estimation in Section 4.4. The image likelihood formulation is presented in Section 4.5. Section 4.6 gives multiple view approach. Section 4.7 provides details on the implementation. The experimental results are presented in Section 4.8. Conclusions and possible directions for future works are given in Section 4.9.

4.2 Related work

There is a significant amount of work on single person tracking and pose estimation. Estimating the 3D pose of multiple people is more challenging than single person tracking and pose estimation, as people may occlude each other, and it is difficult to handle large occlusions efficiently and accurately. We will give a brief overview of related literature discussing three aspects: body model, pose tracking, and occlusion estimation.

4.2.1 Body model

In general, approaches for people tracking and pose estimation can be divided into two categories: a model-free approach and a model-based approach. Model-free approaches do not explicitly use a prior model of the relative configuration of body parts [30, 37, 38]. These approaches first detect individual body parts and then assemble them together into a configuration which best fits observations. However, the search of possible part configurations is quite time consuming. A model-based approach uses an explicit model of a person’s kinematics, shape, appearance, etc. Model-based approaches have been widely used for body pose tracking in video because they provide a detailed description of a human body in 2D or 3D [27, 32, 39, 41, 42]. A 2D model directly relates the model to image features, but depth information can not be exploited due to view limitation. However, depth information is crucial for handling occlusions. Therefore, in order to deal with large occlusions between persons, a 3D model is more suitable [27, 41, 43]. 3D models used are skeleton models, cylinder models, mesh models etc. In general, 3D models give a better representation of the
human body than 2D models resulting in a better accuracy. Although 2D models can be directly used for calculating 2D image features, they need to adapt to different view angles. In this paper, in order to better handle occlusions, we use an articulated 3D body model to represent a human. As for different views, this 3D model is projected onto corresponding image planes. In each of the views, 2D image features are used to calculate the probability of this 3D model.

4.2.2 Pose tracking

To achieve robust visual tracking, particle filtering has been widely used because inferring the exact MAP (Maximum A Posteriori Probability) solution is intractable. When the state space dimensionality is not very high, tracking can be realized in real-time with particle filters. In [27, 89], an annealed particle filtering approach is used to track articulated body motion: an annealing parameter is added to the particle filter. The simulated annealing tends to concentrate samples into one mode. The final results show that the annealed particle filter gives better performance than standard Condensation [63]. However, the improvement is only done in the sampling strategy, i.e., how to evolve and generate samples. The high-dimensional search space is still the same, since all the separate parts of an articulated model are put into one state vector. There is no hierarchical search used in the approach. In [41], an approach is proposed that employs smoothing filters to improve particle-filtered estimations. However, the results show that existing smoothing methods are unable to provide much improvement, due to the high-dimensional nature of body tracking. Therefore, reducing the search dimensionality of a particle filter is crucial for improving tracking accuracy and speeding up systems. In [29], a hierarchical stochastic sampling scheme is proposed for dynamic estimation of poses. It allows a parallel searching by partitioning state parameters into layers. However, certain layers still need to be associated with others, while they could be separated if given a part-based model. In [15], a weak model is used to progressively reduce the search space for human pose estimation so that in the reduced search space a strong model has better chances of finding detailed body part positions. The disadvantage is that the approach needs to train an upper-body detector using manually annotated data. In [106], an approach is proposed to recover human 2D postures from monocular image sequences. A hierarchical 2D body model is employed to constraint body partitions. One of the key applications of the approach is to initialize human body trackers. Therefore, there is no pose tracking integrated in the approach. In this paper, we use a part-based hierarchical model to track body parts in a hierarchical way.

4.2.3 Occlusion estimation

As for multiple people pose estimation, occlusion is one of the main issues to consider. In general, there are two types of occlusions in 2D images, self occlusion and inter-person occlusions (shown in Figure 4.1). Self occlusion can be handled within each individual by using edge and color features [32, 35]. Inter-person occlusion handling needs to take into account physical relationships between persons, such as distance and depth information. In [35], an approach is proposed for automatic initialization and tracking of human poses. The approach combines bottom-up evidence with top-down priors to get efficient pose estimation. The proposed approach can deal with self occlusions by using an appearance
model and an occlusion map. However, the runtime is around 45 seconds per frame due to
the search of possible part configurations. Therefore, it is not suitable for real-time
applications. Moreover, inter-person occlusions are not considered. In [32], a multilevel
structured model is used to track human pose in a coarse to fine way. Partial occlusion
between persons is handled with the appearance model of each person. However, occlusion
handling is limited to monocular sequences. Moreover, the computation speed is on average
5 minutes. In [46, 97] the inter-person occlusion is analyzed using world-to-image projection
and different persons are distinguished by using rich appearance descriptors. Although the
tracking system operates in real-time, there is no further body pose estimation for each
individual. In [107], multiple-viewpoint and viewpoint selection mechanism is used to
reduce self-occlusion and hand-hand occlusion problems. The system integrates information
from all “non-occluded” images to track 3D hand motions and also reduces computation
costs. In order to deal with self occlusions and severe inter-person occlusions efficiently, a
combined global and local occlusion estimation is proposed in this paper.

4.3 Part-based upper body model

The proposed method combines a 3D model with 2D image features. The generative 3D
body model is shown in Figure 4.2. It utilizes six cylinders to represent a person’s head,
torso, right upper arm, right lower arm, left upper arm and left lower arm. The size of each
cylinder is set during initialization and is fixed during tracking. The relative 3D locations of
each cylinder are derived from human body configuration. According to human kinematic
model, we allow torso, head and upper arm to have 3 degrees of freedom (3DOF), that is the
rotation in \(x, y \) and \(z \) directions respectively, and lower arms with 1DOF, the rotation in \(y \)
direction.

Together with \(x, y \) and \(z \) translation of the torso, there are in total 17 parameters \(x_1 \ldots x_{17} \)
in our 3D upper body model (in [89] a whole body model was used with 29 DOF). Assuming
there is a single person at time \(t \), all the parameters of the person can be put into one state
vector: \(X_t = \{x_1 \ldots x_{17}\} \). If there are multiple people in the scene, they are represented by \(X_{t1}, \ldots, X_{tM}. M \) is the total number of persons in the scene. The number of persons is fixed
during tracking and we do not consider person entering or leaving the scene.

A similar upper body model has been used previously for both 2D and 3D pose
estimation [27, 40, 43, 89]. The main difference between our approach and previous
approaches is that our model is hierarchical. Instead of concatenating all parameters into a
single state vector \(X_t \), we split the parameters into three state vectors:

\[
X_{t,1} = \{x_1 \ldots x_9\}, \quad X_{t,2} = \{x_{10} \ldots x_{13}\}, \quad X_{t,3} = \{x_{14} \ldots x_{17}\} \tag{4.1}
\]

\(X_{t,1} \) represents the state of the torso and the head: 3 translation and 3 rotation parameters for
the torso \(x_1 \ldots x_6 \), and 3 rotation parameters for the head \(x_7 \ldots x_9 \), \(X_{t,2} \) represents the state of right
arm, 3 rotation parameters for the upper arm \(x_{10} \ldots x_{12} \), and 1 rotation parameter for the lower
arm \(x_{13} \), \(X_{t,3} \) represents the state of left arm \(x_{14} \ldots x_{17} \), with the same meaning as right arm.
Figure 4.1: Images with self occlusions and inter-person occlusions from different views and different videos. Self occlusion regions are marked with yellow lines. Inter-person occlusion regions are marked with red lines. In all these images, only upper body occlusion regions are marked.

Figure 4.2: A 3D upper body model. (a) In 2D image, head and torso is indicated with rectangles. Arms are represented by skeletons. (b) A 3D upper body model with all the parameters.
4.4. Particle filtering for tracking and pose estimation in a hierarchical way

After we define the body model, our goal is to find body configuration, which can be formulated as a problem of state estimation. We first consider one image sequence. Given a sequence of image observations \( I_{1:t} = (I_1, \ldots, I_t) \), from time 1 to \( t \), the posterior probability of state vector \( X_t \) is denoted as \( p(X_t | I_{1:t}) \). In order to simplify inference, we assume that the process is first order Markovian. There are several papers that use the same assumption [43,
90]. Assuming that the observation is only dependent on the current state, a recursive Bayes formula can be derived and used for inference [105]:

\[
p(X_t | I_t) \propto p(I_t | X_t) \int p(X_t | X_{t-1}) p(X_{t-1} | I_{t-1}) d X_{t-1}
\]

(4.2)

\(p(I_t | X_t)\) is the image likelihood function. The calculation of \(p(I_t | X_t)\) will be discussed in Section 4.5.4. \(p(X_t | X_{t-1})\) is the temporal dynamic model. Since movements of a person are not very predictable and we do not want to limit the temporal dynamic model to certain type of motion, \(p(X_t | X_{t-1})\) is modeled as a Gaussian distribution \(X_t \sim N(X_{t-1}, \Sigma)\), where \(\Sigma\) is the covariance matrix of the dynamic model. A standard method for representing the posterior probability \(p(X_t | I_t)\) is particle filtering.

4.4.1 Particle filtering

Particle filtering is based on sampling from the posterior and it is an iterative process. Particle filtering can be extended to successive images in a video sequence for object tracking [63]. We represent the state density as a set of particles with normalized weights \(\{X_{t,1}^{(i)}, \pi_{t,1}^{(i)}\}_{i=1}^{N_{c,1}}, N_{c,1}\) is the number of particles. Here we only give the estimation of \(X_{t,1}\). The estimation of \(X_{t,2}\) and \(X_{t,3}\) is similar to that of \(X_{t,1}\).

\[
\{X_{t,1}^{(i)}, \pi_{t,1}^{(i)}\}_{i=1}^{N_{c,1}}, \pi_{t,1}^{(i)} = \frac{p(I_t | X_{t,1}^{(i)})}{\sum_{j=1}^{N_{c,1}} p(I_t | X_{t,1}^{(j)})} \quad i \in \{1, \ldots, N_{c,1}\}
\]

(4.3)

An estimation of \(X_{t,1}\) at time \(t\) is selected from state candidates by maximizing a posterior:

\[
\hat{X}_{t,1}^{MAP} = X_{t,1}^{(j)}, \quad j = \arg \max_{i} \pi_{t,1}^{(i)}
\]

(4.4)

In order to get a better estimation of the posterior distribution, a variation of the standard particle filtering is proposed in [27], called annealed particle filtering. It adds several iterations within each time instant. For each iteration (within the same time instant), an annealing parameter is introduced to control the evolution of the sample set. In Section 4.8.2, we compare the performance of standard particle filtering with annealed particle filtering.

4.4.2 Hierarchical search

A hierarchical search is used to localize different parts of the 3D upper body model. We start with searching the torso/head configuration while leaving the arms out, or keeping the arms at their predicted values [96]. After the torso is located, the search is moved to the arms. For
each person, there are three individual particle filters used to estimate state vectors $X_{t,1}$, $X_{t,2}$, and $X_{t,3}$ respectively. State vector $X_{t,1}$ is estimated first. The image likelihood function $p(I_t | X_{t,1})$ is given in Section 4.5.4. After evaluating all the state candidates $X_{t,1}^{(i)}, i=1...N_{t,1}$, the maximum a posteriori poses $\hat{X}_{t,1}^{MAP}$ is computed from the set of particles $\{X_{t,1}^{(i)}, \pi_{t,1}^{(i)}\}_{i=1}^{N_{t,1}}$.

After estimating state vector $X_{t,1}$, state candidates $X_{t,2}^{(i)}, X_{t,3}^{(i)}$ are generated and estimated in parallel. The image likelihood functions $p(I_t | X_{t,2})$ and $p(I_t | X_{t,3})$ can be found in Section 4.5.4. When all the state candidates $X_{t,2}^{(i)}$ and $X_{t,3}^{(i)}$ are evaluated, the maximum a posteriori poses, $\hat{X}_{t,2}^{MAP}$ and $\hat{X}_{t,3}^{MAP}$ are selected from the set of particles $\{X_{t,2}^{(i)}, \pi_{t,2}^{(i)}\}_{i=1}^{N_{t,2}}$ and $\{X_{t,3}^{(i)}, \pi_{t,3}^{(i)}\}_{i=1}^{N_{t,3}}$.

### 4.5 Image likelihood formulation

In this section, we discuss the formulation of image likelihood functions $p(I_t | X_{t,1})$, $p(I_t | X_{t,2})$ and $p(I_t | X_{t,3})$. The image likelihood indicates how well a given pose fits on image observation. Many image features, such as shape, edge, color, can be used or combined to derive image likelihood functions [18, 21, 79, 90, 101-103]. In our implementation, we use foreground silhouette, gradient orientation and appearance to form a combined image likelihood function. We will discuss these features in the next subsections.

#### 4.5.1 Foreground silhouette

Motion is one of the important visual cues to find “interesting objects” in the scene. There are several approaches that can be used to extract moving objects from background, such as frame differencing, kernel density estimation, mixture of Gaussian [80, 87, 98, 104], etc. Frame differencing is simple and fast to implement since it uses a fixed background image. However, it can not adapt to changes in the scene and is very sensitive to noise. Kernel density based methods estimate the probability density function of background or foreground with kernels [87, 104]. The kernels are usually chosen as Gaussian kernels. However, computation complexity of kernel density estimation is quite high. A mixture of Gaussian uses multiple Gaussian functions to model each pixel in the scene [80, 98]. It can deal with changing light conditions and is robust to cluttered background. The processing time can be real time. Therefore, we use a mixture of Gaussians to extract foreground binary image. If a pixel at location $(i, j)$ is considered as a foreground pixel, the pixel value is assigned with 1, otherwise with 0, and a foreground binary image $I^f$ is obtained.

$$I^f(i, j) = \begin{cases} 1 & \text{if } (i, j) \text{ is foreground pixel} \\ 0 & \text{else} \end{cases}$$  (4.5)
Chapter 4. Multiple People Tracking and Pose Estimation with Occlusion Estimation

Figure 4.4: (a) Foreground binary image. (b) Edge image. (c) A template used for calculating edge matching error.

With the foreground binary image $I'$ (Figure 4.4 (a)), we can calculate the foreground matching error $e_f$ of a body part, similar to [89].

$$e_f = \frac{1}{N_p} \sum_{k=1}^{N_p} (1 - I'(i^{(k)}, j^{(k)}))^2$$

(4.6)

$(i^{(k)}, j^{(k)})$ is the $k^{th}$ pixel location in a projected image patch of a body part, given pose $X_t$. $k$ is the pixel index, $N_p$ is the total number of sampled pixels within the image patch. For the torso and the head, projected 2D image patches are approximately 2D rectangles. Therefore an integral image can be used to calculate the $e_f$ of $X_{t,1}$ [81]. In some situations, e.g. when a person is bending, projected 2D image patches for the torso and the head are rotated rectangles. This precludes the use of integral images, but we only validate our approach for vertically standing people.

4.5.2 Gradient orientation

In order to deal with partial and self occlusion, edges are often used to separate different body parts [35, 89]. Since each body part is modeled as a cylinder, the projected body part on an image will consist of parallel edges. For those pixels on these parallel edges, their gradient orientation should be perpendicular to the direction of parallel edges. Therefore instead of using an edge image (Figure 4.4 (b)) or gradient magnitude, we use gradient orientation to calculate the edge matching error $e_e$ of projected image patches.

$$e_e = \frac{1}{N_p} \sum_{k=1}^{N_p} (1 - I^e(i^{(k)}, j^{(k)}))^2$$

(4.7)

$(i^{(k)}, j^{(k)})$ is the $k^{th}$ pixel location in edge regions of the projected image patch (edge regions are the shaded points in Figure 4.4 (c), induced by $X_t$). $k$ is the pixel index, $N_p$ is the total
number of pixels in edge regions. \( I^k(i^{(k)}, j^{(k)}) \) gives the inner product of the normalized gradient orientation vector of the \( k^{th} \) pixel and the normalized vector perpendicular to parallel edges.

### 4.5.3 Appearance

Appearance is also a commonly used feature to identify objects [32, 35, 39, 90]. The appearance model of an object is usually built in normalized RGB color space, or HSV space to exclude the influence of light or shadows. The appearance matching error \( e_a \) of a body part is calculated as follows:

\[
e_a = \left(1 - I^a(c, p)\right)^2
\]  

\( I^a(c, p) \) is the appearance similarity between a projected candidate image patch \( c \) and the modeled image patch \( p \). \( c \) is induced by \( X_t \). The value of \( I^a \) is between 0 and 1. The value 0 means totally mismatch while the value 1 means totally match. The appearance of different body parts can be modeled by a color histogram or a color vector as in [35]. The color histogram gives better representation of color distribution within a body part, but it requires more computation time compared with the color vector. The color vector is easy to compute but less accurate. In our approach, we use color histograms to model the torso appearance. Since the accuracy of torso tracking will directly influence arm tracking, by using color histograms we improve tracking accuracy at little computational costs (Table 4.1).

For torso modeling, the original image is first transferred to normalized RGB color space. Then \( r, g \) components are used to construct color histograms in order to reduce light influence. The color histogram of a torso candidate and the modeled torso are represented as \( H_c, H_p \) respectively. We use the Bhattacharyya coefficient [87] to measure the appearance similarity \( I^a(c, p) \) between a torso candidate and the modeled torso.

\[
I^a(c, p) = \sum_{i=1}^{N_b} \sqrt{H_c(i) \cdot H_p(i)}
\]  

\( H_c(i) \) and \( H_p(i) \) are the normalized histogram value of the \( i^{th} \) bin. \( N_b \) is the total number of bins in the histogram.

As for arm appearance modeling, the color vector is used. The color vector is composed of \( r \) and \( g \) component in normalized RGB color space [35]. Here an upper arm or a lower arm is evenly segmented into \( N_r \) regions along the middle line (Figure 4.5). The value of \( N_r \) depends on image resolution and body part size. For each of the \( N_r \) regions, a single color value is used to represent the appearance of this region. The color vector of the modeled arm is described as \( V_p \):

\[
V_p = (r^p_1, g^p_1, r^p_2, g^p_2, ..., r^p_{N_r}, g^p_{N_r})
\]
$r_i^p, g_i^p$ are the mean r and g component of the $i^{th}$ part, respectively. In our experiments, we set $N_r$ to 10. Since the size of candidate image patches for upper arms and lower arms is much smaller than the torso, using color vector representation gives comparable accuracy as color histograms (Table 4.2). However, calculating a color vector is faster than a color histogram. Specifically, when the number of candidate image patches is large, this advantage becomes important for a real time application. The appearance similarity $I^a(c, p)$ between an arm candidate and the modeled arm is:

$$I^a(c, p) = 1 - \frac{1}{N_r} \sum_{i=1}^{N_r} \sqrt{ (r_i^c - r_i^p)^2 + (g_i^c - g_i^p)^2 }$$  \hspace{1cm} (4.11)$$

### 4.5.4 Combined image likelihood

The image features, foreground silhouette, gradient orientation and appearance are combined together to estimate the image likelihood functions $p(I_t|X_{t,i}), i=1,2,3$. $i$ selects the state vector.

$$p(I_t|X_{t,i}) \propto \exp\left[ -(\lambda_f e_f + \lambda_e e_e + \lambda_a e_a) \right]$$ \hspace{1cm} (4.12)$$

The weighting factors $\lambda_f, \lambda_e$ and $\lambda_a$ are determined empirically and are currently fixed. From the experiments, we found that the foreground silhouette is a more reliable feature than gradient orientation and appearance. The reason is that background subtraction works quite well in most of our test videos. Therefore, we have relatively good foreground silhouettes in the test videos. As for the appearance, if people wear different color of clothes, the appearance is a very good feature to distinguish them. However, if people’s clothes have the same color (such as in Test Video 4), the appearance does not help much. As for the gradient orientation, it might be influenced by clothes’ texture and cluttered background (such as in Test Video 2). In general, the foreground silhouette is a more reliable feature than the gradient orientation and appearance. In our current approach, we therefore give the foreground silhouette a higher weight than the gradient orientation and appearance. In future work, we will investigate how to set the weighting factors adaptively to the scene.
4.6 Multiple views approach

4.6.1 Multiple views combining

If there are multiple views, the image likelihood functions are estimated in a similar way:

$$p(I_1^v I_2^v \ldots I_N^v | X_{i,j}) \propto \exp \left[ - \sum_{j=1}^{N} \left( \lambda_j e_j^v + \lambda_e e_e^v + \lambda_a e_a^v \right) \right]$$  (4.13)

$N_v$ is the number of views and $j$ is the view index. When there are multiple people in the scene, we can simply track them separately. The image likelihood functions for person $k$ are as follows:

$$p^{(k)}(I_1^v I_2^v \ldots I_N^v | X_{i,j}) \propto \exp \left[ - \sum_{j=1}^{N} \left( \lambda_j e_j^{(k)} + \lambda_e e_e^{(k)} + \lambda_a e_a^{(k)} \right) \right]$$  (4.14)

$k$ is the index of the person in the scene ($k=1 \ldots M$, $M$ is the total number of persons in the scene).

4.6.2 Global occlusion estimation

In order to calculate the image likelihood functions, the 3D human body is projected onto 2D views and the depth information is lost. Therefore when persons are close to each other in 3D, there may appear occlusions between persons in 2D views. When inter-person occlusion happens in a certain projected 2D view, some image features like edges and appearance within this view may change. Therefore inter-person occlusion will influence image likelihood and cause tracking failure. In order to deal with inter-person occlusions, we propose a novel approach of using global occlusion estimation to modify the image likelihood functions.

$$p^{(k)}(I_1^v I_2^v \ldots I_N^v | X_{i,j}) \propto \exp \left[ - \sum_{j=1}^{N} \left( \alpha g_j^v \cdot (\lambda_j e_j^{(k)} + \lambda_e e_e^{(k)} + \lambda_a e_a^{(k)}) \right) \right]$$  (4.15)

$g_j^v$ is the global visibility map of person $k$, in view $j$. $\alpha$ is a constant set manually. $g_j^v$ is defined as follows:

$$g_j^v = \min \left[ 1, \min_{r \neq k} \left( V_j(k,r) \right) \right]$$  (4.16)

$$V_j(k,r) = \begin{cases} \frac{D_j(k,r)}{T_j(k,r)} & \text{if } P_j(k) > P_j(r) \\ 1 & \text{else} \end{cases}$$  (4.17)
$D_j(k, r)$ is the Euclidean distance between the projected 2D torso center position of person $k$ and person $r$, at time $t-1$. $T_j(k, r)$ is a distance threshold that guarantees that there is no inter-person occlusions between person $k$ and person $r$. $P_j(k)$ is the distance between the torso of person $k$ and camera $j$. $P_j(r)$ is the distance between the torso of person $r$ and camera $j$. The value of $T_j(k, r)$ depends on the rotation and the size/scale of person $k$ and person $r$. Therefore we use projected torso width in view $j$ to calculation $T_j(k, r)$:

$$T_j(k, r) = 0.5 \times [W_j(k) + W_j(r)] \tag{4.18}$$

$W_j(k), W_j(r)$ are the projected torso width of person $k$ and person $r$, in view $j$. The value of $g_j^l$ is between 0 and 1. If the persons are close to each other, the value of $g_j^l$ becomes smaller and vice versa. $g_j^l$ represents a global occlusion estimation between persons by integrating previously estimated poses and locations.

The global visibility map is used to update image likelihood functions of each individual, as shown in Figure 4.6. Image features from different views are weighted according to the view visibility. Multiple views are combined in such a way that we rely more on the features derived from reliable views and less on those from occluded views. In this way, the proposed approach reduces the confusion caused by the occluded view and improves the tracking accuracy.

The proposed global visibility map is a person level visibility estimation. Our emphasis is put more on efficiently view combining, rather than on occlusion reasoning like the pixel level occlusion approach [35]. Moreover, the person level occlusion approach gives a global estimation for the scene and is less noisy compared with a pixel level occlusion approach. Our approach is particularly suitable for recordings with a larger number of views ($N_v > 2$), since in such cases, the pixel level occlusion approach can be very time consuming. Obviously, the number of cameras needed for robust tracking is also related to the number of persons to be tracked.

### 4.6.3 Local occlusion estimation

Besides inter-person occlusion, we also take into account self occlusions in the scene. For instance, a person’s arm may occlude or be occluded by the person’s torso. In this step we do not consider a person’s part occluding or being occluded by other person’s body parts, since the global occlusion estimation handles this inter-person occlusion. In order to deal with self occlusions, a local visibility map $l_j^i$ is used to calculate image likelihood functions:

$$p^{(k)}(I_1^j I_2^j \ldots I_{N_v}^j | X_{t-1}) \propto \exp \left[ -\sum_{j=1}^{N_v} (g_k^j)^\alpha \cdot (l_j^i)^\beta \cdot \left( \hat{\lambda}_j e_f^{(j,k)} + \hat{\lambda}_c e_c^{(j,k)} + \hat{\lambda}_a e_a^{(j,k)} \right) \right] \tag{4.19}$$

$$l_j^i(i) = \frac{C_j(i)}{\sum_{j=1}^{N_v} C_j(i)} \tag{4.20}$$
4.7 Implementation

4.7.1 Experiment setup

In order to test the proposed approach, we recorded four video sequences at three different environments. Since so far there is no benchmark sequence available for multiple people.
pose estimation with ground truth data, the evaluation was done with our own recorded video sequences. The test video sequences consist of over 5000 frames at a resolution of $640 \times 480$ pixels and at a frame rate of 25fps. People are allowed to have large and fast movement of their arms and whole body. Test Video 1 was recorded in the motion lab of Utrecht University. The ground truth of Test Video 1 was obtained with motion capture data. Therefore we can compare tracking accuracy of our proposed approach with ground truth in 3D. There are two persons in this video. Test Video 2 is the video sequence recorded in a meeting room, with cluttered background. Since we do not have the motion capture data of Test Video 2, we manually labeled joint positions in 2D images and compare tracking accuracy of Test Video 2 in 2D. Test Video 3 was recorded in the same environment as Test Video 1, but with three persons in the scene. In Test Video 1, 2 and 3, people are continuously performing 9 poses defined in [88]. These poses are designed to control games. It also explains why people do not have large turns in these videos, since they are supposed to face the screen. In order to test the robustness of the proposed approach, we also recorded a video with natural movement, Test Video 4. Test Video 4 consists of walking people making turns and waving their hands. All of these four video sequences are available upon request. Besides the four recorded video sequences, we also tested our approach on one of the sequences from EPFL data set (http://cvlab.epfl.ch/data/pom/). In preliminary experiments, we also evaluated our approach on the PETS2009 data set. However, we found that this data set is not suitable for our experiments because camera views are not synchronized. Test Video 1, 2 and 3 were recorded with three cameras. Test Video 4 was recorded with four cameras. The cameras are installed at the locations shown in Figure 4.7 to cover an approximate area of 4m x 5m. For each of these views, background subtraction is used to obtain foreground binary images. In the background subtraction implementation, the number of the Gaussian models is chosen to be 3. In order to exclude shadows from foreground images, a shadow removing approach is employed [98].

4.7.2 Initialization step

In the initialization, the system automatically detects different body parts by using an initial pose. The initial pose is defined as people stretching both of their arms sideway. The initial pose is shown in Figure 4.2 (a). From the prior knowledge of this initial pose and general geometrical properties of human beings, shoulders, hands and elbows positions are extracted in the first frame and used for segmenting different body parts. It is assumed that there is no occlusion between persons during initialization.

When people are appearing in the scene, the projected 2D upper body model is used to first find the person’s head and torso. From the expected torso position, we can roughly locate the person’s shoulders. Then a simple skin color model [19] is utilized to detect the person’s hands on foreground images. When the distance between two hands is close to the height of the foreground blob and stays constant for several frames, the person is assumed performing the initial pose. In this case, the full arm length ($FAL$) can be obtained, being the Euclidean distance between the shoulder $(x_s, y_s, z_s)$ and the hand $(x_h, y_h, z_h)$ on the same side of the body. The full arm length is equal to:

$$FAL = \sqrt{(x_s - x_h)^2 + (y_s - y_h)^2 + (z_s - z_h)^2} \quad (4.21)$$
4.8 Experimental results

4.8.1 Pose tracking results

Some pose tracking results of our approach are shown in Figure 4.8. From the results, we see that the proposed method can estimate multiple persons’ poses simultaneously and correctly. Occlusions between persons in one or two views are correctly handled with information from other non-occluded views. Although in some frames one of the persons is totally occluded by the other, he/she is still being correctly tracked. Test Video 2 is a more challenging video due to the cluttered background and the clothing of the person. However, the proposed method also gives robust tracking results (for more results, please see our submitted video data).

4.8.2 Quantitative comparison

We quantitatively compared our method with ground truth. The evaluated joints are head center, right shoulder, right elbow, right wrist, left shoulder, left elbow and left wrist. The joint error (in mm) is used to quantitatively measure the accuracy and is defined as follows:

\[
(x_e, y_e, z_e) = \text{mean}[(x_h, y_h, z_h), (x_s, y_s, z_s)]
\]
\[
Error = \sqrt{(x_j - x_g)^2 + (y_j - y_g)^2 + (z_j - z_g)^2}
\] (4.23)

where \((x_j, y_j, z_j)\) is the tracked joint position, \((x_g, y_g, z_g)\) is the ground truth.

Table 4.3 shows the pose tracking results of Test Video 1 with the hierarchical particle filtering (HPF) and with the hierarchical annealed particle filtering (HAPF) (The average and the standard deviation of the error are given). Test Video 1 consists of 1500 frames. The error is averaged over all 1500 frames. The results show that the performance of the HPF and the HAPF are comparable for most joints. The selected sample images (Figure 4.9) show that the right arm of P1 is missed with the HPF, while it is correctly tracked with the HAPF. However this difference only happens in a few frames among the whole image sequence. In general, the average performance of the HPF and the HAPF is very close (Table 4.3). Since we track different body parts in a hierarchical way, the high-dimensional search space is broken into lower dimensional subspace. In these lower dimensional subspaces, the risk for the tracking system running into local minimum is already reduced by hierarchical searching. In this way, the advantage of using annealed particle filtering is not that obvious any more. The boxplots of the error with the HAPF are shown in Figure 4.10.

We compared the performance of our approach with the annealed particle filtering (APF) without using hierarchical search. Table 4.4 gives a quantitative comparison of our approach and the APF using the average and the standard deviation of the error (in mm). Here we use only a part of Test Video 1 as test video, since tracking is lost after a certain time by using the APF. Figure 4.11 shows some results obtained with the APF. For non-occlusion frames (shown in Figure 4.11 (a)), different body parts still can be tracked, but with less accuracy compared with our proposed approach. If there are inter-person occlusions in the view (shown in Figure 4.11 (b)), the confusion between persons causes the APF failure. Since in the APF all the parameters for different body parts are put into one state vector, the miss matching of one body part will directly influence the matching of other body parts.

For comparison, we conducted two other tests with Test Video 4. Test Video 4 consists of 1000 frames. During the first 600 frames, people are just standing still. There is not much motion within the first 600 frames. Therefore we exclude them from evaluation. We manually labelled the ground truth of frame# 620 to frame# 720. We use these frames to evaluate our approach. In the first test, the global and local occlusion estimation is not used. Table 4.5 shows the average and the standard deviation of the tracking error (in pixel) with and without occlusion estimation. By using the global and the local occlusion estimation, the tracking error of individuals and their body joints is reduced significantly. In the second test, the number of views is varied. Figure 4.12 depicts the graph of average joint error. It shows the advantage of multiple views for tracking persons under severe occlusions. It indicates that with \(Nv > 2\) the tracking error is decreased dramatically. For stable tracking, the minimum required view number is three for Test Video 4. We also analyzed the frames in which some body parts are missed. Some tracking failure examples are shown in Figure 4.13. In Test Video 1, there are less tracking failures compared with Test Video 2. The reason is that the clothing in Test Video 2 causes more confusion than in Test Video 1. In Test Video 2, the two person’s clothes share similar colors. Additionally, one of them is wearing clothes with quite some textures on it. In order to show the movement of different body parts, we plot the trajectory of one person’s hand from frame 400 to 800. Here we use the Test Video 2 as an example and the trajectory is represented by 2D pixel coordinates.
4.8. Experimental results

Figure 4.8: Pose estimation results of different videos with the proposed approach (for more results, please see submitted video data). Each row shows the multiple views of the same time frame. Each column shows different frames in time. Person’s head and torso are approximated by rectangles. Person’s arms are represented by skeletons. (a) Sample images from Test Video 1. (b) Sample images from Test Video 2. (c) Sample images from Test Video 3. (d) Sample images from Test Video 4.

Figure 4.14 shows the tracked hand position with manually labeled position. The red line in Figure 4.14 is the manually labeled $x$ and $y$ coordinate of the right hand of P2. The blue line is the tracked right hand coordinates of P2 by our proposed approach. It can be seen that although the person’s hand has large and fast movement in both $x$ and $y$ direction during tracking, the proposed approach gives robust tracking results. Figure 4.15 shows the multiple people tracking results of “Terrace sequences” from the EPFL data set. In this test, we track three persons in the scene. Since our approach handles fixed number of tracked persons, we do not considering people entering/leaving the scene. However, new people entering in the scene (frame# 880, frame# 956) did not influence our tracker to track correctly on the true persons. We also compared our tracking results with the ground truth. The average tracking errors for the three persons are 12.3cm, 19.6cm, and 12.2cm. The result is quite accurate, given that the actual grid resolution is 20cm in ground truth. As for pose estimation, we can see that people’s arms are not well located. This is due to the low resolution of the sequences. Our approach requires image resolution of which a person’s height is at least 200 pixels.
Chapter 4. Multiple People Tracking and Pose Estimation with Occlusion Estimation

Figure 4.9: Pose estimation results with the HPF and the HAPF. (a) The right arm of P1 (the occluded person) is missed with the HPF. (b) The right arm of P1 is correctly tracked with the HAPF.

Figure 4.10: Pose tracking results of the proposed method. (a) Boxplots of the joint error of Person1 (P1) in Test Video 1. (b) Boxplots of the joint error of Person2 (P2) in Test Video 1. Joint index 1 to 7 indicates head center, right shoulder, right elbow, right wrist, left shoulder, left elbow, and left wrist. The error is given in mm.
Figure 4.11: Illustration of pose estimation results using the APF. (a) Frames without inter-person occlusions. (b) Frames with inter-person and self occlusions. For non-occlusion frames (Figure 4.11 (a)), different body parts still can be tracked, but with less accuracy compared with our proposed approach (Table 4.4). If there are inter-person occlusions in the view (Figure 4.11 (b)), the confusion between persons causes the APF failure.

Figure 4.12: Pose tracking results of the proposed method with different number of views of Test Video 4. The error is given in pixel and is the average error of all the joints compared with ground truth.
Chapter 4. Multiple People Tracking and Pose Estimation with Occlusion Estimation

Figure 4.13: Some tracking failure examples of our proposed approach.

Figure 4.14: Comparison of the tracked hand position with manually labeled ground truth of Test Video 2. (a) x coordinate of the right hand position of P2. (b) y coordinate of the right hand position of P2.
4.8. Experimental results

Figure 4.15: Multiple people tracking results of “Terrace sequences” from EPFL with the proposed approach (for more results, please see submitted video data). Each row shows the multiple views of the same time frame. Each column shows different frames in time. Persons’ head and torso are approximately with rectangles. Person’s arms are represented by skeletons.

Table 4.1: Pose tracking error with and without using color histogram for torso tracking. The error is given in pixel. The average error is denoted by (Ave.) and the standard deviation is denoted by (Std.). The data set is Test Video 2.

<table>
<thead>
<tr>
<th>Body joints</th>
<th>With using color histogram</th>
<th>Without using color histogram</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ave.</td>
<td>Std.</td>
</tr>
<tr>
<td>Torso center</td>
<td>10.01</td>
<td>3.30</td>
</tr>
<tr>
<td>Right shoulder</td>
<td>11.87</td>
<td>5.60</td>
</tr>
<tr>
<td>Right elbow</td>
<td>13.65</td>
<td>6.89</td>
</tr>
<tr>
<td>Right wrist</td>
<td>17.38</td>
<td>9.64</td>
</tr>
<tr>
<td>Left shoulder</td>
<td>10.62</td>
<td>6.23</td>
</tr>
<tr>
<td>Left elbow</td>
<td>12.99</td>
<td>7.47</td>
</tr>
<tr>
<td>Left wrist</td>
<td>14.21</td>
<td>10.30</td>
</tr>
</tbody>
</table>

Table 4.1: Pose tracking error with and without using color histogram for torso tracking. The error is given in pixel. The average error is denoted by (Ave.) and the standard deviation is denoted by (Std.). The data set is Test Video 2.
Table 4.2: Arm tracking error and speed comparison with using color histogram and using color vector. The error is given in pixel. The average error is denoted by (Ave.) and the standard deviation is denoted by (Std.). The data set is Test Video 2. The speed is given in second/frame.

<table>
<thead>
<tr>
<th>Body joints</th>
<th>Arm tracking using color vector</th>
<th>Arm tracking using color histogram</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ave.</td>
<td>Std.</td>
</tr>
<tr>
<td>Right elbow</td>
<td>14.59</td>
<td>9.50</td>
</tr>
<tr>
<td>Right wrist</td>
<td>18.41</td>
<td>11.31</td>
</tr>
<tr>
<td>Left elbow</td>
<td>12.72</td>
<td>7.04</td>
</tr>
<tr>
<td>Left wrist</td>
<td>16.75</td>
<td>11.77</td>
</tr>
<tr>
<td>Speed</td>
<td>≈0.005</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.3: Pose tracking error with the HPF and the HAPF. The error is given in mm. The average error is denoted by (Ave.) and the standard deviation is denoted by (Std.). The standard deviation is given in the brackets. P1 is one of the two persons in Test Video 1 and he is labeled with red lines in Figure 4.8(a). P2 is the other person in Test Video 1 and he is labeled with green lines in Figure 4.8(a).

<table>
<thead>
<tr>
<th>Body joints</th>
<th>Ave.(Std.) with the HPF</th>
<th>Ave.(Std.) with the HAPF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P1</td>
<td>P2</td>
</tr>
<tr>
<td>Head center</td>
<td>51.8(17.1)</td>
<td>24.8(8.2)</td>
</tr>
<tr>
<td>Right shoulder</td>
<td>70.4(13.6)</td>
<td>28.1(14.5)</td>
</tr>
<tr>
<td>Right elbow</td>
<td>79.5(49.1)</td>
<td>109.1(60.4)</td>
</tr>
<tr>
<td>Right wrist</td>
<td>137.1(67.0)</td>
<td>181.7(128.3)</td>
</tr>
<tr>
<td>Left shoulder</td>
<td>79.0(34.2)</td>
<td>18.1(8.0)</td>
</tr>
<tr>
<td>Left elbow</td>
<td>60.6(35.3)</td>
<td>75.1(32.7)</td>
</tr>
<tr>
<td>Left wrist</td>
<td>108.1(67.3)</td>
<td>128.8(67.2)</td>
</tr>
</tbody>
</table>

Table 4.4: Pose tracking error with our approach and the APF. The error is given in mm. The average error is denoted by (Ave.) and the standard deviation is denoted by (Std.). The data set is Test Video 1.

<table>
<thead>
<tr>
<th>Body joints</th>
<th>Our approach</th>
<th>The APF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ave.</td>
<td>Std.</td>
</tr>
<tr>
<td>Head center</td>
<td>25.9</td>
<td>7.8</td>
</tr>
<tr>
<td>Right shoulder</td>
<td>29.9</td>
<td>13.2</td>
</tr>
<tr>
<td>Right elbow</td>
<td>118.0</td>
<td>64.2</td>
</tr>
<tr>
<td>Right wrist</td>
<td>139.8</td>
<td>85.9</td>
</tr>
<tr>
<td>Left shoulder</td>
<td>21.7</td>
<td>7.5</td>
</tr>
<tr>
<td>Left elbow</td>
<td>74.8</td>
<td>39.3</td>
</tr>
<tr>
<td>Left wrist</td>
<td>121.7</td>
<td>73.8</td>
</tr>
</tbody>
</table>
4.8.3 Computation

The processing speed of our current implementation is 10 to 13 frames per second using a 2.66GHz Core Duo PC. With further code optimization and usage of graphics hardware, we believe that the speed still can be increased with code optimization.

4.9 Conclusions and future works

In this paper, we presented an approach to track multiple people simultaneously and estimate their 3D pose in a hierarchical way. In order to deal with inter-person occlusions, we proposed a global occlusion estimation approach that combines the information from multiple views. The global visibility map of each individual in all the views is calculated and used to weigh image observation from those views. The self occlusion is handled by the local occlusion estimation. We have performed experiments on several challenging video sequences and successfully tracked multiple people’ poses. In particular, the results suggest that the combination of global and local occlusion estimation results in significant improvement in system performance regarding to the tracking accuracy.

Another feature of our approach is that a part-based hierarchical model is used to track upper body poses. We first estimated the relatively easy detectable body parts, such as the head and the torso. Then from the roughly located shoulder positions, we started searching for the left arm and the right arm in parallel. Given the constraints of human kinematic model, the finding of each body part is independent. Therefore our method reduces the search space dimensionality and enables a hierarchical search. The hierarchical way of search reduces the computational complexity required to track the given part-based model, compared with alternatives like annealed particle filtering. The speed up of the system is necessary for many real-time and online applications.

This work can be extended into tracking a small group of people and full body pose estimation. The future research direction is how to extract more robust image features to
form better image observations and therefore further reduce confusions between persons, or between body parts. Further we aim at extending pose estimation into pose recognition to be used for recognizing interactions between persons.

Acknowledgments

This research has been supported by the GATE (Game Research for Training and Entertainment) project, funded by the Netherlands Organization for Scientific Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie). Thank reviewers for their input and thank Laurens van der Maaten for proofreading this paper.
Chapter 5

Human Interaction Recognition through Multiple People Pose Tracking

Abstract

This paper presents and investigates a general approach for the recognition of human interactions from video data. The proposed approach directly uses extracted low-level features to recognize high-level human interactions. A frame-based approach is investigated to represent different interactions, which incorporates the spatial and temporal structure of each interaction. In order to find out the most informative features to distinguish these interactions, we analyze the classifier performance on different feature spaces. The experimental results show that the most informative features to separate desired interactions is the 3D joint positions which represent the spatial structure of each interaction. Motion information, which represents the temporal structure of each interaction, helps to distinguish certain classes, but might also decrease the classification performance of other classes due to overtraining.

This chapter has been submitted to the Image and Vision Computing journal as the paper “Human interaction representation and recognition through multiple people pose tracking” by F. Huo, D.M.J. Tax, and E.A. Hendriks.
5.1 Introduction

Human interaction representation and recognition is a key technology for many computer-vision based applications, such as surveillance, human-computer interactions and serious games [64, 108]. The challenge of human interaction representation and recognition is to extract reliable features from images or video content and further use these features to represent and recognize different interactions, such as two-person interactions [54, 69, 70, 73], group activities [52, 71, 74-76], and social interactions [77], etc. In general, recognition performances highly depend on reliable feature extraction. There are many computer-vision based methods which can be used to extract different visual features from human interaction contents. The features can be spatial features: such as interest points [68, 111], shape [76], and appearance [70, 114]; temporal features: such as human motion features [86, 110, 115], optical flow [54], etc. The choice among these features depends on the available video data types as well as the video content.

Most of the paper put focus on extracting features and building classifiers to achieve the best classification performance [54, 109, 116]. Less attention has been paid on investigating what are the most informative features to represent certain types of interaction: spatial features, or temporal features, or the combination of both. In this paper, we mainly focus on analyzing the influence of different feature spaces on the classifier performance. Our feature spaces are composed of 3D joint positions obtained from our previous tracking system [112]. The 3D-based tracking systems are view-independent and have the advantage of analyzing the details of human movements [64]. With the development of stereo camera systems and motion sensing devices such as kinect, more and more research is applied to 3D human motion analysis [117, 118].

Human activity recognition methods can be divided into two categories: single level approaches and hierarchical approaches [64]. Single level approaches [54, 55, 68] directly use the extracted features for human activity recognition without any intermediate processes. Such approaches have been successfully used in recognizing individual activities with simple motion patterns, such as walking, running, and jumping. Hierarchical approaches [69-72] are more often used for representing and recognizing complex human activities and interactions. Hierarchical approaches describe human activities at multiple levels. For instance, low level models individual body part motion; middle level models single person actions; and high level models two-person interactions. The major drawback of hierarchical approaches is that errors might be generated at any of the intermediate steps and these errors propagate through the following steps. The more steps between low level models and high level models, the larger the chance that new errors might be generated, decreasing the overall performance. Moreover, for other interactions, it is difficult to generally define the required intermediate steps. Single level approaches can also be used in complex human activity recognitions, having the advantage of being more efficient, less prone to cumulative errors and more generic than hierarchical approaches, but comes usually with the price of higher dimensional feature spaces.

In this paper, we propose a single level general method which directly uses tracked 3D joint positions of two interacting persons to recognize two-person interactions: shake hands, introduce, point, punch, wave and push (Example images are shown in Figure 5.1). However,
the proposed approach is not limited to these predefined interactions. Both spatial and temporal features are used to represent each interaction. The emphasis of this paper is on investigating what are the most informative features to distinguish these interactions while keeping good recognition performance. In this paper, we use two people interaction recognition as an example. However, this work can be easily extended into multiple people (more than two) interaction recognition, if their 3D joint positions are available. Our previous tracking system is able to track up to three persons simultaneously, even with self and inter-person occlusions [112]. Using the 3D joint positions of each pair of persons, multiple people interaction recognition can be realized.

The main contributions of our work are as follows:

1. A frame-based representation approach is proposed to describe different interactions.

   The proposed approach does not require the exact time duration of different interactions, circumventing the problem of unknown-time-duration interactions representation and recognition. We do include temporal/ motion information into this frame-based approach. Therefore, the proposed approach is able to distinguish interactions with different spatial and temporal structures.

2. Investigation of classifier performance on different feature spaces which combine different spatial and temporal structures.
We compare different ways of representing spatial and temporal information for the purpose of interaction recognition. The experimental results show that the most informative features to separate desired interactions are the 3D joint positions which represent the spatial structure of each interaction. Adding motion information helps to distinguish certain classes, but might also deteriorate the classification performance of other classes due to overtraining.

The paper is organized as follows. A brief overview of related work is given in Section 5.2. We discuss multiple people 3D pose tracking in Section 5.3 and discuss human interaction representation and recognition in Section 5.4. Section 5.5 gives experimental results. Conclusions and future work are given in Section 5.6.

5.2 Related work

We discuss related work from three aspects: multiple people motion tracking, feature extraction and activities representation and recognition.

5.2.1 Multiple people motion tracking

There has been a significant amount of work carried out on single person motion tracking, but not many on multiple people tracking. Tracking 3D motions of multiple people is more challenging than that of single persons due to occlusion problems. In [86], an approach is proposed to track 3D poses of multiple people. The proposed method uses an appearance model and an occlusion map to handle self occlusions. However, inter-person occlusions are not considered. In [110], an appearance-based approach is proposed to track multiple body parts of interacting humans under the conditions of occlusion and shadow. Individual pixels in the color image sequence are first grouped into homogeneous blobs and further into body parts. In this approach occlusion handling and tolerance is limited to monocular sequences. In our previous work, we proposed a multiple view approach to track 3D poses of multiple people [112]. A combined global and local occlusion estimation is proposed to deal with self occlusions and as well as inter-person occlusions. In this paper, this multiple people pose tracking approach is used to extract motion features for human interaction representation and recognition.

5.2.2 Feature extraction

Instead of extracting human motions from video sequences, other features, such as local features, corners, and interest points, can also be used for activity recognition [54, 55, 68, 111]. In [111], feature selection and classification are combined into a single framework. First an over-complete set of simple 2D Harris corners are extracted from video sequences. Then a hierarchical approach is proposed to iteratively group these 2D corners spatially and temporally. At each hierarchical stage, data mining is used to learn distinctive feature models. At the end, the learned feature model is used for action recognition in test sequences. The proposed approach gives real-time performance on video data [111]. Since one activity recorded from different view angles has different 2D features, the learned feature model is also different. The proposed method is view dependent. In [54], a person-centered descriptor
is introduced to represent each person detected in a frame. The descriptor, which is composed of histograms of gradients and optical flow, is used as a feature vector for classifiers training. The experimental results show that with different combinations of descriptor parameters, the classification accuracy also differs. For each interaction, the best combination of parameters is desired, which makes the proposed descriptor not very general.

5.2.3 Activities representation and recognition

Approaches for recognizing human interactions can be classified into three categories: description-based approaches [69], statistical approaches [52, 53, 71, 72] and classification-based approaches [54-56]. In [69], a description-based approach is proposed for reliably recognizing human actions and interactions. Based on the recognition of low-level poses and gestures, the system hierarchically recognizes high-level actions and interactions. The framework integrates a probabilistic decision into a description-based approach. However, the drawback of this description-based approach is that human experts have to encode human activities into sub-events in a particular order. It requires a large amount of manually labeling and it is quite subjective. Moreover, the time interval between the sub-events is hard to define. In [72], a layered hidden Markov model (LHMM) representation is used to model human activities in a hierarchical manner. The proposed LHMMs can be considered as a cascade of HMMs. Therefore, it is feasible to train each level of the hierarchy independently. They demonstrated that the accuracy of LHMMs is significantly higher than that of single, standard HMMs, given the same amount of training data. Moreover, the proposed LHMMs are more robust to environment changes than HMMs in an office-awareness application. However, there is no further discussion about the possible extension to other human interaction applications. In [54], a classification-based approach is used to recognize interactions in TV shows. An initial set of linear support-vector-machine (SVM) classifiers are trained for each interaction. Further on, the classification performance is improved by combining head orientation and people location in a frame. Although motion features are calculated by optical flow and used as one of the classification features, long-term motion information is not considered in this paper.

The aim of our paper is to investigate an approach that circumvents some of the problems mentioned above by using a simple generic frame-based way of representing spatial and temporal information. Instead of 2D positions we will use 3D positions by using multiple cameras to be viewpoint independent. The objective is to classify each frame into one of the learned classes including also a non-interaction class. The classes are learned beforehand.

5.3 Multiple people 3D pose tracking

Before we can explain the proposed interaction representation, we first explain the representation of one 3D pose. In [112] a method is proposed to track multiple people simultaneously and estimate their pose in 3D. This model emphasizes on tracking of the upper body and arms, since in general upper body poses are more frequently used in human-
computer-interaction (HCI) applications. The output of the tracker are 3D joint positions. The extracted joints are torso center \((tc)\), head center \((hc)\), right shoulder \((rs)\), right elbow \((re)\), right wrist \((rw)\), left shoulder \((ls)\), left elbow \((le)\), and left wrist \((lw)\). The 3D positions of these joints in a predefined world coordinate system are presented by \([x_i, y_i, z_i]^T\), \(i \in \{tc, hc, ..., lw\}\) (A visual representation is given in Figure 5.2). The joint positions of multiple people are presented as \([x_i^{(k)}, y_i^{(k)}, z_i^{(k)}]^T\), \(i \in \{tc, hc, ..., lw\}\), \(k\) is the total number of tracked persons.

This algorithm is applied to interaction videos that have been recorded in an indoor environment. Some results are shown in Figure 5.3. Each row shows multiple views of the same time instant. Each column shows different interaction sequences. A person’s head and torso are approximated with rectangles in 2D images. A person’s arms are represented by skeletons.

### 5.4 Human interaction representation and recognition

#### 5.4.1 Human interaction representation

We directly use the extracted 3D joint positions \([x_i^{(k)}, y_i^{(k)}, z_i^{(k)}]^T\) to represent human interactions. The feature vector \(F_i\) for current frame \(t\) is composed of 3D joint positions of the two interacting persons. Since each interaction is determined by Person1 and as well as Person2, both of the two persons’ joint positions are needed for representing the interaction. Because there are 8 upper body joints for each individual, it means that for two interacting persons, the total joint number is 16. The \(x\), \(y\), and \(z\) coordinates of these 16 joints are concatenated into one feature vector \(F_i\) (Figure 5.4). Although there are two ways to concatenate the two persons’ joint positions, the order does not make any difference for the recognition. In our training data as well as test data, Person1 may “point” at Person2 and vice versa. In both cases, they belong to the interaction class: “point”. We also did experiments of changing the order and the results showed no difference at the classification error. However, if the task is to recognize who is pointing at whom, the order will make a difference, but we did not try to solve this problem in this research.

#### 5.4.2 Spatial feature vector

We use four feature vectors \(F_i^{(1)}\), \(F_i^{(2)}\), \(F_i^{(3)}\), and \(F_i^{(4)}\) to represent persons’ pose. These feature vectors describe the spatial structure of the two persons’ poses by means of different scaling and normalizing the 3D joint positions. The motivation of choosing these feature vectors is given in the following paragraphs.

Feature vector \(F_i^{(1)}\) (Figure 5.4 (a)) is the concatenation of the 3D joint positions of Person1 and Person2. The 3D world origin is located at Person1’s torso center. The \(x\) direction of the world coordinate is the frontal direction of Person1 and \(z\) direction is perpendicular to the ground floor. The disadvantage of using the feature vector \(F_i^{(1)}\) is that \(F_i^{(1)}\) does not represent the tree structure of a human body explicitly. In a tree structure, a person’s pose is explicitly described by using the relative 3D joint positions. Therefore we propose to use another feature vector \(F_i^{(2)}\).
5.4. Human interaction representation and recognition

Figure 5.2: 3D upper body model used in body parts tracking. The extracted joints are head center \((hc)\), torso center \((tc)\), right shoulder \((rs)\), right elbow \((re)\), right wrist \((rw)\), left shoulder \((ls)\), left elbow \((le)\), and left wrist \((lw)\).

Figure 5.3: Pose estimation results of different interaction videos with our previously proposed approach. Each row shows multiple views of the same time instant. Each column shows different interaction sequences.
Figure 5.4: Feature vector $F_t$ composed of two persons’ 3D joint positions in frame $t$. (a) Absolute positions: feature vector $F_t^{(1)}$. (b) Relative positions: feature vector $F_t^{(2)}$. (c) Fixed normalized relative positions: feature vector $F_t^{(3)}$. (d) Time dependent normalized relative positions: feature vector $F_t^{(4)}$.

The 3D world coordinate of the feature vector $F_t^{(2)}$ (Figure 5.4 (b)) is the same as that of the feature vector $F_t^{(1)}$. Except for the torso position, the remain joint positions are all relative 3D positions to the corresponding 3D torso position. The relative joint positions describe persons’ poses while the torso positions indicate persons’ locations in the scene. Compared with the feature vector $F_t^{(1)}$, the feature vector $F_t^{(2)}$ represents persons’ pose and position more explicitly.

For the same interaction, people may perform it at different locations. Since in the feature vector $F_t^{(1)}$ and $F_t^{(2)}$, the torso position is one of the components, the classification performance may be influenced by the (arbitrary) torso positions. In order to exclude this influence, for $F_t^{(3)}$ (Figure 5.4 (c)), we normalize the 3D world coordinate in such a way that Person1’s torso located at $(0, 0, 0)$, person2’s torso located at $(\tilde{x}_c^{(2)}, 0, 0)$, $[\tilde{x}_c^{(2)}, \tilde{y}_c^{(2)}, \tilde{z}_c^{(2)}]^T = R [x_c^{(2)}, y_c^{(2)}, z_c^{(2)}]^T$, where $R$ is a $3 \times 3$ rotation matrix. Further on, all the joint positions are divided by a fixed scale $s$. $s = \tilde{x}_c^{(2)}$, $\tilde{x}_c^{(2)}$ is the distance between Person1 and Person2 in the first frame ($t = 1$). In this way, Person1’s start position is normalized at $(0, 0, 0)$ while Person2’s start position is normalized at $(1, 0, 0)$. When $t > 1$, Person2’s torso position will change.

Instead of using a fixed scale, this scale parameter can be adjusted according to the
distance between the two persons, excluding the influence of torso position differences over time. Therefore, feature vector \( F_{t}^{(4)} \) (Figure 5.4 (d)) is computed with a variable scale, depending on the position of Person2 in each frame. First the 3D world coordinate is rotated in the same way as feature vector \( F_{t}^{(3)} \) so that Person1’s torso is at \((0, 0, 0)\) and Person2’s torso is at \((\tilde{x}_w^{(2)}, 0, 0)\). Then joint positions are divided by a variable scale \( s' \), where \( s' = \tilde{x}_w^{(2)} \), \( \tilde{x}_w^{(2)} \) is the distance between Person1 and Person2 in each frame. In this way, Person1’s torso is at \((0, 0, 0)\) and Person2’s torso is at \((1, 0, 0)\) for all the frames \((t \geq 1)\).

### 5.4.3 Temporal feature vector

So far the constructed feature vectors only consist of 3D joint positions representing the spatial structure of each interaction in each frame. Since interactions are dynamic sequences, motion features can well be used to represent the temporal structure of each interaction. Motion features are obtained from a sequence of previous frames \( t-1, t-2, \ldots, t-\Delta t \), where \( \Delta t \) is the number of previous frames from which we use motion information. In total, we construct 5 different types of motion vectors: \( M^{(0)}, M^{(1)}, M^{(2)}, M^{(3)} \) and \( M^{(4)} \) (Figure 5.5).

Motion vector \( M^{(0)} \) just concatenates the joint positions from previous frames, \( M^{(0)} = \{ F_{t-1}, F_{t-2}, \ldots, F_{t-\Delta t} \} \). It is the easiest and simplest way to add motion information from previous frames. A disadvantage is that the dimensionality of \( M^{(0)} \) increases proportionally with increasing \( \Delta t \). Considering the number of samples in the dataset, this high dimensional feature space may cause the classifier to fail.

An alternative way to obtain motion information is to use the joint position difference between frames. Motion vector \( M^{(1)} = \{ F_{t-1} - F_{t-1}, F_{t-2} - F_{t-2}, \ldots, F_{t-\Delta t} - F_{t-\Delta t} \} \) (Figure 5.5(a)), is composed of the position difference of the current frame and previous frames. This representation suffers equally from the increase of dimensionality as \( M^{(0)} \). Both the motion vector \( M^{(0)} \) and \( M^{(1)} \) end up in high dimensionality when \( \Delta t \) is large. In a high dimensional feature space, a large number of training samples are needed. However, in most situations, we can not have sufficient samples. Therefore, alternative motion vectors \( M^{(2)}, M^{(3)} \) and \( M^{(4)} \) are defined with lower dimensionality. This is done by selecting or averaging over previous frames. However, this averaging can be done in different ways. We have chosen two different ways to measure averaged motion information.

Motion vector \( M^{(2)} \) (Figure 5.5 (b)) only consists of position difference between the current frame \( t \) and frame \( t-\Delta t \), \( M^{(2)} = \{ F_{t} - F_{t-\Delta t} \} \). Therefore, the dimension of motion vector \( M^{(2)} \) is fixed (48D) independent of \( \Delta t \). A disadvantage of using motion vector \( M^{(2)} \) is that only part of the motion information is taken into account.

Motion vector \( M^{(3)} \) (Figure 5.5 (c)) uses the average motion information over previous frames. The average position difference between current frame \( t \) and frame \( t-1, t-2, \ldots, t-\Delta t \) is calculated, \( M^{(3)} = \frac{1}{\Delta t} \sum_{i=1}^{\Delta t} \{ F_{t-i} - F_{t-i} \} \). In this case, the dimensionality of the feature vector stays 48D with increasing \( \Delta t \). However \( M^{(3)} \) might be very sensitive to the current joint positions \( F_{t} \). If \( F_{t} \) is not accurate, \( M^{(3)} \) can be very noisy. Therefore, an alternative way is proposed to calculate the averaged motion information as follows.
Figure 5.5: For different values of Δt, different motion information is added to a feature vector. The x-axis (T) indicates frames at different times. The maximum number of previous frames we use is 10, based on the averaged length of the interaction. (a) Motion feature $M^{(1)}$. The dimensionality of a feature vector increases proportionally with the increasing of Δt. (b) Motion feature $M^{(2)}$. The dimensionality of a feature vector stays the same. (c) Motion feature $M^{(3)}$. The dimensionality of a feature vector stays the same. (d) Motion feature $M^{(4)}$. The dimensionality of a feature vector stays the same.

Motion vector $M^{(4)}$ (Figure 5.5 (d)) estimates the average motion information over previous frames, by averaging single frame differences, $M^{(4)} = \frac{1}{\Delta t} \sum_{i=0}^{\Delta t-1} \{F_{i+1} - F_{i+1-1}\}$. Although the single frame differences may be noisy, averaging may result in more reliable motion estimate. There are of course other ways to calculate a motion pattern from a motion trajectory, but these are more or less similar to one of those described above. Therefore in this paper, we only focus on the aforementioned five motion vectors.

5.4.4 Human interaction recognition

We built a classifier for each desired interaction class (Figure 5.1: shake hands, introduce, point, punch, wave and push) and a non-interaction class. The classification is done per frame. So each frame in a sequence is assigned to a class. In total, we have a set of seven one-vs-the-rest classifiers which model $\hat{p}(\omega_i | \bar{x}), i = 1, 2, \ldots, 7$. $\hat{p}(\omega_i | \bar{x})$ is the posterior probability that an input feature vector $\bar{x}$ belongs to a certain class. $\omega_i$ are the trained
5.5. Experimental results

5.5.1 Experiment setup

In order to test the proposed approach, we recorded 6 different two-person interactions: shake hands, introduce, point, punch, wave, and push. Each of the interaction is performed by 8 different pairs of people. Each pair performs each interaction at least three times. People are allowed to stand at different locations in the scene as long as their upper-body are visible in most of the views. For each interaction, people can perform it with different speed each time and within a different time duration. We use four cameras for recording. The four cameras are installed at the locations shown in Figure 5.7 to cover an approximate area of 2.5m×3m. The video sequences are recorded at a resolution of 640×480 pixels and a frame rate of 25fps.

5.5.2 Dataset

For each of the recorded video sequences, we manually labeled the start frame and end frame of each interaction. All the other frames between the start frame and end frame are considered as positive samples of the interaction. Some example images are shown in Figure 5.8. The number of frames for each interaction is shown in Table 5.1. We also give the average length of each interaction by frame numbers. The dataset is available upon request.
5.5.3 Classification results

5.5.3.1 Comparison of several classifier performances

First, we compared the performances of several classifiers with different complexity, including linear and non-linear classifiers. Specifically, we evaluated the least absolute shrinkage and selection operator (Lasso) [113], the linear discriminant analysis (LDA) classifier and the quadratic analysis (QDA) classifier. The feature space is now fixed to \( F_t^{(i)} + M_t^{(b)} \), with \( 0 \leq \Delta t \leq 10 \) (based on the averaged length of the interaction, the maximum number of previous frames we use is 10). Principal component analysis (PCA) is used to reduce the feature space dimensionality for LDA and QDA, keeping 99% variance of the data.

Since there are eight pairs of persons in our dataset, we did an eight-fold cross-validation. For each fold, we leave out the dataset of one pair of persons and use it as a test set. The remaining dataset of the seven pairs of persons are used as a training set. The average classification error of the eight-fold cross-validation is used as the final classification error. We also calculate the standard deviation of the error over the cross-validations. The class prior for positive samples and negative samples is set to 1/7 and 6/7 respectively, for both training and test set. The cross-validation results are shown in Tables 5.2-5.4. The performance of different classifiers differs for each class. In general, PCA+QDA has the best performance compared with Lasso and PCA+LDA. The results show that most of the interactions can be well classified with an error lower than 10%. The classification error of the “point” and the “punch” classes are relatively larger compared with other classes. The reason is that the “point” class and the “punch” class share similar poses as well as motion trajectories. The confusion between these two classes causes the relative large error.

Figure 5.7: The recording camera locations mapped on the ground floor.
5.5. Experimental results

![Figure 5.8: Some examples images from our dataset. Each row shows the frames (from one of the four views) of one interaction. Each column shows different interaction sequences. Person’s head and torso are approximately with rectangles. Person’s arms are represented by skeletons.](image)

Tables 5.2-5.4 also show that the classification error of only using 3D joint positions (when $\Delta t = 0$) and the classification error of combining 3D joint positions with (basic) motion information (when $\Delta t \geq 1$) are very close to each other. The maximum error difference is 4%. Apparently, the most informative features to separate the defined interactions are the 3D joint positions in the current frame, representing the current spatial structure of the interaction. Surprisingly, adding motion information (temporal structure of the interaction) does not improve the overall classification performance. The reason is that the 3D joint positions are sufficient to distinguish the defined interactions from their spatial structures. For instance, the “push” class is well separated from other classes without using any temporal information (Tables 5.2-5.4). In this case, temporal features are not essential any more. When the spatial features are not informative enough to separate the different interactions, temporal features can play a vital role. For example, the “point” and the “punch” classes share similar spatial structures. Adding temporal information does help to reduce the classification error of these two classes (Figure 5.10).

### 5.5.3.2 Comparison of different feature spaces

Next, we compare the performance of PCA+QDA using different feature and motion vectors. In Tables 5.5-5.8, the average classification error over all the seven classes is given. The
Figure 5.9: The average classification error over all the 7 classes of using PCA+QDA. (a) Feature space $F_i^{(1)} + M_j^{(j)}$, (b) Feature space $F_i^{(2)} + M_j^{(j)}$, (c) Feature space $F_i^{(3)} + M_j^{(j)}$, (d) Feature space $F_i^{(4)} + M_j^{(j)}$. The standard deviation of the error is between 0.02 and 0.04.

errors are also plotted in Figure 5.9. The results show that for different combination of $F_i^{(i)}$, $i \in \{1,2,3,4\}$ with $M_j^{(j)}$, $j \in \{0,1,2,3,4\}$, the average classification error does not change dramatically. The classifiers perform slightly worse on feature space $F_i^{(2)} + M_j^{(j)}$ compared with $F_i^{(1)} + M_j^{(j)}$, $F_i^{(3)} + M_j^{(j)}$ and $F_i^{(4)} + M_j^{(j)}$. In feature space $F_i^{(2)} + M_j^{(j)}$, the 3D joint positions (except torso center) are the relative 3D positions to torso center. Apparently these relative positions can not represent different interactions as well as the original 3D joint positions. From Figure 5.9, we see that the combination of $F_i^{(1)} + M_j^{(0)}$ gives the best performance. We plot the error of each class on feature space $F_i^{(1)} + M_j^{(0)}$ in Figure 5.10. For different classes, the influence of $\Delta t$ is different. For instance, when $\Delta t$ is large than 5, the classification error of the non-interaction class increases considerably. The non-interaction class is composed of frames that do not contain the defined 6 interactions. Therefore there is
no uniform motion pattern in the non-interaction class. With adding more motion information, it will not help improve the classification performance and even increase the classification error due to over training. It also shows in Figure 5.10 that the classification error of the “shake hands” and “push” classes is already quite small even without using any motion information. It is because that the pose contained in these two classes is clearly different from other classes. This also explains why the “shake hands” and “push” classes outperform over others. From the overall performance of the six interaction classes, we see that adding motion vectors does not provide additional information over the position features. The classifier performance is mainly determined by the 3D joint positions, representing the spatial structure of the interaction. However, motion information does help to improve classification performance when two classes share similar spatial structures, but with different temporal structures, such the “point” and “punch” classes in Figure 5.10.

Since there is no 3D benchmark dataset for human interaction recognition, the comparison is done with our own datasets (Figure 5.8). There are papers working on a similar interaction recognition problem, but with 2D information only [49, 70]. Because we
Table 5.1: The number of frames for each interaction.

<table>
<thead>
<tr>
<th>class</th>
<th>frames/samples</th>
<th>average length (number of frames)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 shake hand</td>
<td>487</td>
<td>20</td>
</tr>
<tr>
<td>2 introduce</td>
<td>852</td>
<td>22</td>
</tr>
<tr>
<td>3 point</td>
<td>761</td>
<td>21</td>
</tr>
<tr>
<td>4 punch</td>
<td>307</td>
<td>17</td>
</tr>
<tr>
<td>5 wave</td>
<td>521</td>
<td>30</td>
</tr>
<tr>
<td>6 push</td>
<td>652</td>
<td>24</td>
</tr>
<tr>
<td>7 non-interaction</td>
<td>717</td>
<td>11</td>
</tr>
</tbody>
</table>

Table 5.2: Cross-validation results of Lasso classifier on $F_i^{(1)} + M^{(0)}$ feature space. The average classification error of eight-fold cross-validation is given. The standard deviation of the error is given in brackets.

<table>
<thead>
<tr>
<th>$\Delta t$</th>
<th>shake hands</th>
<th>introduce</th>
<th>point</th>
<th>punch</th>
<th>wave</th>
<th>push</th>
<th>non-interaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.10(0.13)</td>
<td>0.09(0.07)</td>
<td>0.12(0.05)</td>
<td>0.15(0.03)</td>
<td>0.05(0.04)</td>
<td>0.03(0.02)</td>
<td>0.08(0.04)</td>
</tr>
<tr>
<td>1</td>
<td>0.10(0.14)</td>
<td>0.09(0.07)</td>
<td>0.12(0.05)</td>
<td>0.15(0.04)</td>
<td>0.05(0.04)</td>
<td>0.03(0.02)</td>
<td>0.07(0.02)</td>
</tr>
<tr>
<td>2</td>
<td>0.09(0.13)</td>
<td>0.09(0.07)</td>
<td>0.11(0.05)</td>
<td>0.15(0.05)</td>
<td>0.05(0.05)</td>
<td>0.03(0.02)</td>
<td>0.06(0.02)</td>
</tr>
<tr>
<td>3</td>
<td>0.10(0.13)</td>
<td>0.11(0.12)</td>
<td>0.11(0.05)</td>
<td>0.14(0.04)</td>
<td>0.05(0.05)</td>
<td>0.03(0.02)</td>
<td>0.06(0.03)</td>
</tr>
<tr>
<td>4</td>
<td>0.09(0.13)</td>
<td>0.10(0.08)</td>
<td>0.11(0.04)</td>
<td>0.14(0.04)</td>
<td>0.05(0.05)</td>
<td>0.02(0.02)</td>
<td>0.06(0.03)</td>
</tr>
<tr>
<td>5</td>
<td>0.08(0.13)</td>
<td>0.12(0.12)</td>
<td>0.11(0.05)</td>
<td>0.14(0.05)</td>
<td>0.05(0.06)</td>
<td>0.02(0.02)</td>
<td>0.07(0.03)</td>
</tr>
<tr>
<td>6</td>
<td>0.08(0.13)</td>
<td>0.13(0.15)</td>
<td>0.12(0.05)</td>
<td>0.15(0.07)</td>
<td>0.05(0.06)</td>
<td>0.02(0.03)</td>
<td>0.07(0.03)</td>
</tr>
<tr>
<td>7</td>
<td>0.07(0.13)</td>
<td>0.12(0.15)</td>
<td>0.10(0.05)</td>
<td>0.14(0.05)</td>
<td>0.05(0.06)</td>
<td>0.02(0.02)</td>
<td>0.08(0.03)</td>
</tr>
<tr>
<td>8</td>
<td>0.07(0.14)</td>
<td>0.11(0.08)</td>
<td>0.11(0.06)</td>
<td>0.13(0.03)</td>
<td>0.06(0.06)</td>
<td>0.02(0.02)</td>
<td>0.08(0.03)</td>
</tr>
<tr>
<td>9</td>
<td>0.07(0.14)</td>
<td>0.10(0.07)</td>
<td>0.11(0.07)</td>
<td>0.13(0.03)</td>
<td>0.05(0.06)</td>
<td>0.02(0.03)</td>
<td>0.10(0.03)</td>
</tr>
<tr>
<td>10</td>
<td>0.07(0.15)</td>
<td>0.10(0.08)</td>
<td>0.10(0.06)</td>
<td>0.13(0.04)</td>
<td>0.06(0.06)</td>
<td>0.01(0.02)</td>
<td>0.11(0.03)</td>
</tr>
</tbody>
</table>

have 3D information, more features can be extracted from the 3D data. The emphasis of our paper, however, is not on recognition accuracy, but on investigating what are the most informative features to distinguish the interactions while keeping good recognition performance. Therefore, we did not compare with other existing action recognition algorithms quantitatively. The necessity of informative feature lies in practical applications. By using the informative features, even a linear classifier (LDA) is shown to be sufficient for interaction recognition. This is quite attractive from a computational point of view.
5.5. Experimental results

Table 5.3: Cross-validation results of PCA+LDA classifier on $F_i^{(1)} + M_i^{(0)}$ feature space. The average classification error of eight-fold cross-validation is given. The standard deviation of the error is given in brackets.

<table>
<thead>
<tr>
<th>$\Delta t$</th>
<th>shake hands</th>
<th>introduce</th>
<th>point</th>
<th>punch</th>
<th>wave</th>
<th>push</th>
<th>non-interaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.08(0.10)</td>
<td>0.17(0.08)</td>
<td>0.11(0.05)</td>
<td>0.15(0.07)</td>
<td>0.07(0.07)</td>
<td>0.04(0.05)</td>
<td>0.08(0.05)</td>
</tr>
<tr>
<td>1</td>
<td>0.08(0.10)</td>
<td>0.17(0.08)</td>
<td>0.11(0.05)</td>
<td>0.14(0.07)</td>
<td>0.06(0.07)</td>
<td>0.04(0.06)</td>
<td>0.08(0.05)</td>
</tr>
<tr>
<td>2</td>
<td>0.09(0.11)</td>
<td>0.17(0.08)</td>
<td>0.11(0.05)</td>
<td>0.14(0.06)</td>
<td>0.05(0.07)</td>
<td>0.04(0.05)</td>
<td>0.08(0.05)</td>
</tr>
<tr>
<td>3</td>
<td>0.07(0.11)</td>
<td>0.17(0.08)</td>
<td>0.11(0.05)</td>
<td>0.15(0.07)</td>
<td>0.05(0.06)</td>
<td>0.04(0.06)</td>
<td>0.08(0.05)</td>
</tr>
<tr>
<td>4</td>
<td>0.06(0.12)</td>
<td>0.18(0.08)</td>
<td>0.11(0.07)</td>
<td>0.15(0.07)</td>
<td>0.04(0.06)</td>
<td>0.03(0.06)</td>
<td>0.08(0.06)</td>
</tr>
<tr>
<td>5</td>
<td>0.06(0.10)</td>
<td>0.19(0.09)</td>
<td>0.12(0.07)</td>
<td>0.15(0.06)</td>
<td>0.04(0.06)</td>
<td>0.03(0.05)</td>
<td>0.08(0.06)</td>
</tr>
<tr>
<td>6</td>
<td>0.06(0.11)</td>
<td>0.19(0.09)</td>
<td>0.12(0.07)</td>
<td>0.15(0.06)</td>
<td>0.04(0.06)</td>
<td>0.03(0.06)</td>
<td>0.08(0.06)</td>
</tr>
<tr>
<td>7</td>
<td>0.05(0.10)</td>
<td>0.20(0.09)</td>
<td>0.12(0.07)</td>
<td>0.15(0.06)</td>
<td>0.04(0.06)</td>
<td>0.03(0.05)</td>
<td>0.08(0.06)</td>
</tr>
<tr>
<td>8</td>
<td>0.05(0.11)</td>
<td>0.20(0.09)</td>
<td>0.12(0.07)</td>
<td>0.15(0.07)</td>
<td>0.04(0.07)</td>
<td>0.03(0.06)</td>
<td>0.08(0.07)</td>
</tr>
<tr>
<td>9</td>
<td>0.04(0.08)</td>
<td>0.21(0.09)</td>
<td>0.12(0.07)</td>
<td>0.16(0.07)</td>
<td>0.04(0.07)</td>
<td>0.03(0.05)</td>
<td>0.09(0.09)</td>
</tr>
<tr>
<td>10</td>
<td>0.04(0.08)</td>
<td>0.21(0.10)</td>
<td>0.11(0.07)</td>
<td>0.15(0.07)</td>
<td>0.05(0.08)</td>
<td>0.03(0.05)</td>
<td>0.09(0.11)</td>
</tr>
</tbody>
</table>

Table 5.4: Cross-validation results of PCA+QDA classifier on $F_i^{(1)} + M_i^{(0)}$ feature space. The average classification error of eight-fold cross-validation is given. The standard deviation of the error is given in brackets.

<table>
<thead>
<tr>
<th>$\Delta t$</th>
<th>shake hands</th>
<th>introduce</th>
<th>point</th>
<th>punch</th>
<th>wave</th>
<th>push</th>
<th>non-interaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.02(0.05)</td>
<td>0.09(0.06)</td>
<td>0.11(0.06)</td>
<td>0.11(0.04)</td>
<td>0.07(0.05)</td>
<td>0.02(0.03)</td>
<td>0.02(0.01)</td>
</tr>
<tr>
<td>1</td>
<td>0.02(0.05)</td>
<td>0.08(0.05)</td>
<td>0.11(0.06)</td>
<td>0.10(0.05)</td>
<td>0.08(0.05)</td>
<td>0.02(0.03)</td>
<td>0.02(0.01)</td>
</tr>
<tr>
<td>2</td>
<td>0.02(0.05)</td>
<td>0.09(0.06)</td>
<td>0.10(0.06)</td>
<td>0.10(0.06)</td>
<td>0.08(0.06)</td>
<td>0.02(0.03)</td>
<td>0.02(0.01)</td>
</tr>
<tr>
<td>3</td>
<td>0.02(0.05)</td>
<td>0.08(0.06)</td>
<td>0.10(0.06)</td>
<td>0.10(0.06)</td>
<td>0.09(0.06)</td>
<td>0.03(0.05)</td>
<td>0.02(0.02)</td>
</tr>
<tr>
<td>4</td>
<td>0.03(0.05)</td>
<td>0.08(0.06)</td>
<td>0.09(0.06)</td>
<td>0.10(0.05)</td>
<td>0.10(0.06)</td>
<td>0.03(0.06)</td>
<td>0.02(0.02)</td>
</tr>
<tr>
<td>5</td>
<td>0.03(0.05)</td>
<td>0.09(0.06)</td>
<td>0.10(0.06)</td>
<td>0.10(0.06)</td>
<td>0.10(0.06)</td>
<td>0.03(0.06)</td>
<td>0.02(0.02)</td>
</tr>
<tr>
<td>6</td>
<td>0.02(0.05)</td>
<td>0.08(0.06)</td>
<td>0.10(0.06)</td>
<td>0.10(0.06)</td>
<td>0.10(0.06)</td>
<td>0.04(0.06)</td>
<td>0.03(0.02)</td>
</tr>
<tr>
<td>7</td>
<td>0.02(0.05)</td>
<td>0.08(0.06)</td>
<td>0.09(0.06)</td>
<td>0.10(0.07)</td>
<td>0.10(0.06)</td>
<td>0.04(0.06)</td>
<td>0.03(0.02)</td>
</tr>
<tr>
<td>8</td>
<td>0.02(0.05)</td>
<td>0.07(0.06)</td>
<td>0.09(0.06)</td>
<td>0.09(0.07)</td>
<td>0.10(0.06)</td>
<td>0.04(0.06)</td>
<td>0.04(0.02)</td>
</tr>
<tr>
<td>9</td>
<td>0.02(0.05)</td>
<td>0.08(0.06)</td>
<td>0.09(0.06)</td>
<td>0.08(0.05)</td>
<td>0.10(0.06)</td>
<td>0.05(0.06)</td>
<td>0.05(0.03)</td>
</tr>
<tr>
<td>10</td>
<td>0.03(0.05)</td>
<td>0.08(0.06)</td>
<td>0.08(0.06)</td>
<td>0.09(0.05)</td>
<td>0.10(0.06)</td>
<td>0.06(0.07)</td>
<td>0.06(0.05)</td>
</tr>
</tbody>
</table>
Table 5.5: The average classification error over all the 7 classes on $F^1_t$ of using PCA+QDC. The standard deviation of the error is given in brackets.

<table>
<thead>
<tr>
<th>$\Delta t$</th>
<th>$M^{(0)}$</th>
<th>$M^{(1)}$</th>
<th>$M^{(2)}$</th>
<th>$M^{(3)}$</th>
<th>$M^{(4)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.06(0.04)</td>
<td>0.06(0.04)</td>
<td>0.06(0.04)</td>
<td>0.06(0.04)</td>
<td>0.06(0.04)</td>
</tr>
<tr>
<td>1</td>
<td>0.06(0.04)</td>
<td>0.07(0.04)</td>
<td>0.07(0.04)</td>
<td>0.07(0.04)</td>
<td>0.07(0.04)</td>
</tr>
<tr>
<td>2</td>
<td>0.06(0.04)</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.04)</td>
<td>0.07(0.04)</td>
</tr>
<tr>
<td>3</td>
<td>0.06(0.04)</td>
<td>0.07(0.03)</td>
<td>0.06(0.03)</td>
<td>0.07(0.04)</td>
<td>0.06(0.04)</td>
</tr>
<tr>
<td>4</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
<td>0.06(0.03)</td>
<td>0.06(0.04)</td>
<td>0.07(0.04)</td>
</tr>
<tr>
<td>5</td>
<td>0.07(0.04)</td>
<td>0.07(0.03)</td>
<td>0.06(0.03)</td>
<td>0.06(0.04)</td>
<td>0.07(0.04)</td>
</tr>
<tr>
<td>6</td>
<td>0.07(0.03)</td>
<td>0.08(0.03)</td>
<td>0.07(0.03)</td>
<td>0.06(0.03)</td>
<td>0.07(0.04)</td>
</tr>
<tr>
<td>7</td>
<td>0.07(0.03)</td>
<td>0.08(0.02)</td>
<td>0.07(0.03)</td>
<td>0.06(0.03)</td>
<td>0.07(0.04)</td>
</tr>
<tr>
<td>8</td>
<td>0.06(0.03)</td>
<td>0.09(0.02)</td>
<td>0.07(0.03)</td>
<td>0.06(0.03)</td>
<td>0.07(0.04)</td>
</tr>
<tr>
<td>9</td>
<td>0.07(0.04)</td>
<td>0.09(0.02)</td>
<td>0.07(0.03)</td>
<td>0.07(0.04)</td>
<td>0.07(0.04)</td>
</tr>
<tr>
<td>10</td>
<td>0.07(0.02)</td>
<td>0.10(0.02)</td>
<td>0.07(0.03)</td>
<td>0.07(0.04)</td>
<td>0.07(0.04)</td>
</tr>
</tbody>
</table>

Table 5.6: The average classification error over all the 7 classes on $F^2_t$ of using PCA+QDC. The standard deviation of the error is given in brackets.

<table>
<thead>
<tr>
<th>$\Delta t$</th>
<th>$M^{(0)}$</th>
<th>$M^{(1)}$</th>
<th>$M^{(2)}$</th>
<th>$M^{(3)}$</th>
<th>$M^{(4)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
</tr>
<tr>
<td>1</td>
<td>0.07(0.03)</td>
<td>0.08(0.03)</td>
<td>0.08(0.03)</td>
<td>0.08(0.03)</td>
<td>0.08(0.03)</td>
</tr>
<tr>
<td>2</td>
<td>0.07(0.03)</td>
<td>0.08(0.03)</td>
<td>0.08(0.03)</td>
<td>0.08(0.03)</td>
<td>0.07(0.03)</td>
</tr>
<tr>
<td>3</td>
<td>0.08(0.03)</td>
<td>0.09(0.03)</td>
<td>0.08(0.03)</td>
<td>0.08(0.03)</td>
<td>0.08(0.03)</td>
</tr>
<tr>
<td>4</td>
<td>0.08(0.03)</td>
<td>0.09(0.02)</td>
<td>0.08(0.02)</td>
<td>0.08(0.03)</td>
<td>0.08(0.03)</td>
</tr>
<tr>
<td>5</td>
<td>0.08(0.03)</td>
<td>0.09(0.02)</td>
<td>0.08(0.02)</td>
<td>0.08(0.03)</td>
<td>0.08(0.03)</td>
</tr>
<tr>
<td>6</td>
<td>0.09(0.02)</td>
<td>0.10(0.01)</td>
<td>0.08(0.02)</td>
<td>0.08(0.02)</td>
<td>0.08(0.02)</td>
</tr>
<tr>
<td>7</td>
<td>0.09(0.02)</td>
<td>0.10(0.01)</td>
<td>0.08(0.02)</td>
<td>0.08(0.02)</td>
<td>0.08(0.02)</td>
</tr>
<tr>
<td>8</td>
<td>0.09(0.02)</td>
<td>0.11(0.02)</td>
<td>0.09(0.01)</td>
<td>0.08(0.02)</td>
<td>0.08(0.02)</td>
</tr>
<tr>
<td>9</td>
<td>0.10(0.02)</td>
<td>0.12(0.02)</td>
<td>0.09(0.01)</td>
<td>0.08(0.02)</td>
<td>0.08(0.02)</td>
</tr>
<tr>
<td>10</td>
<td>0.10(0.02)</td>
<td>0.13(0.01)</td>
<td>0.09(0.01)</td>
<td>0.09(0.02)</td>
<td>0.09(0.02)</td>
</tr>
</tbody>
</table>
### Table 5.7: The average classification error over all the 7 classes on $F_i^{(3)}$ of using PCA+QDC. The standard deviation of the error is given in brackets.

<table>
<thead>
<tr>
<th>$\Delta t$</th>
<th>$M^{(0)}$</th>
<th>$M^{(1)}$</th>
<th>$M^{(2)}$</th>
<th>$M^{(3)}$</th>
<th>$M^{(4)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.06(0.03)</td>
<td>0.06(0.03)</td>
<td>0.06(0.03)</td>
<td>0.06(0.03)</td>
<td>0.06(0.03)</td>
</tr>
<tr>
<td>1</td>
<td>0.06(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
</tr>
<tr>
<td>2</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
</tr>
<tr>
<td>3</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
</tr>
<tr>
<td>4</td>
<td>0.07(0.03)</td>
<td>0.08(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
<td>0.06(0.03)</td>
</tr>
<tr>
<td>5</td>
<td>0.07(0.03)</td>
<td>0.08(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
</tr>
<tr>
<td>6</td>
<td>0.08(0.03)</td>
<td>0.08(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
</tr>
<tr>
<td>7</td>
<td>0.08(0.03)</td>
<td>0.09(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
</tr>
<tr>
<td>8</td>
<td>0.08(0.03)</td>
<td>0.09(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
</tr>
<tr>
<td>9</td>
<td>0.09(0.03)</td>
<td>0.10(0.04)</td>
<td>0.08(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
</tr>
<tr>
<td>10</td>
<td>0.09(0.03)</td>
<td>0.10(0.04)</td>
<td>0.08(0.03)</td>
<td>0.08(0.03)</td>
<td>0.08(0.03)</td>
</tr>
</tbody>
</table>

### Table 5.8: The average classification error over all the 7 classes on $F_i^{(4)}$ of using PCA+QDC. The standard deviation of the error is given in brackets.

<table>
<thead>
<tr>
<th>$\Delta t$</th>
<th>$M^{(0)}$</th>
<th>$M^{(1)}$</th>
<th>$M^{(2)}$</th>
<th>$M^{(3)}$</th>
<th>$M^{(4)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.07(0.04)</td>
<td>0.07(0.04)</td>
<td>0.07(0.04)</td>
<td>0.07(0.04)</td>
<td>0.07(0.04)</td>
</tr>
<tr>
<td>1</td>
<td>0.07(0.04)</td>
<td>0.07(0.04)</td>
<td>0.07(0.04)</td>
<td>0.07(0.04)</td>
<td>0.07(0.04)</td>
</tr>
<tr>
<td>2</td>
<td>0.07(0.04)</td>
<td>0.07(0.04)</td>
<td>0.07(0.04)</td>
<td>0.07(0.03)</td>
<td>0.07(0.04)</td>
</tr>
<tr>
<td>3</td>
<td>0.07(0.04)</td>
<td>0.08(0.04)</td>
<td>0.07(0.03)</td>
<td>0.07(0.04)</td>
<td>0.07(0.04)</td>
</tr>
<tr>
<td>4</td>
<td>0.07(0.04)</td>
<td>0.08(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.04)</td>
<td>0.07(0.04)</td>
</tr>
<tr>
<td>5</td>
<td>0.07(0.04)</td>
<td>0.08(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.04)</td>
<td>0.07(0.04)</td>
</tr>
<tr>
<td>6</td>
<td>0.08(0.04)</td>
<td>0.08(0.03)</td>
<td>0.07(0.03)</td>
<td>0.07(0.04)</td>
<td>0.07(0.04)</td>
</tr>
<tr>
<td>7</td>
<td>0.08(0.04)</td>
<td>0.09(0.04)</td>
<td>0.07(0.03)</td>
<td>0.07(0.04)</td>
<td>0.07(0.04)</td>
</tr>
<tr>
<td>8</td>
<td>0.08(0.04)</td>
<td>0.09(0.04)</td>
<td>0.07(0.03)</td>
<td>0.07(0.04)</td>
<td>0.07(0.04)</td>
</tr>
<tr>
<td>9</td>
<td>0.09(0.04)</td>
<td>0.10(0.04)</td>
<td>0.08(0.03)</td>
<td>0.08(0.04)</td>
<td>0.07(0.04)</td>
</tr>
<tr>
<td>10</td>
<td>0.09(0.04)</td>
<td>0.10(0.04)</td>
<td>0.08(0.04)</td>
<td>0.08(0.04)</td>
<td>0.08(0.04)</td>
</tr>
</tbody>
</table>
5.6 Conclusions and future works

In this paper we present a general approach for human interaction representation and recognition. It provides an easy solution for unknown-time-duration interactions representation and recognition by incorporating the temporal and spatial structure of each interaction and it does not depend on the specific type of interactions. We analyzed the classifier performance on different feature spaces. The overall classification performance indicates that spatial structure is more informative to separate different interactions, compared to the temporal structure of each class. However, if certain classes have large overlap in their spatial structure, temporal information helps to further separate these classes. This is visible, for instance, in the “point” and the “punch” classes.

In this paper, our emphasis is on two-person interaction recognition. These interactions can be considered as basic elements of group activities. We will further extend two-person interaction recognition into interaction detection from a group of people. The limitation of our approach is that it cannot separate interactions which differ in local features, such as hand shape, gaze direction, etc. By combining 3D joint positions with such local features will definitely improve the classification performance and increase classifiers capability.

Acknowledgments

This research has been supported by the GATE (Game Research for Training and Entertainment) project, funded by the Netherlands Organization for Scientific Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie).
Chapter 6

Discussions and Future Directions

In this thesis, a complete framework for multiple people tracking and interaction recognition is proposed. We aimed at fast solutions for multiple people tracking problem. Therefore, our focus was put on the simplicity and efficiency of the algorithm. We investigated methodologies that can fit well in real applications, such as health care, education, training, serious games, etc. The proposed approaches are able to track up to three persons and estimate their 3D poses simultaneously. The processing speed is 10 to 13 frames per second for upper body tracking, which is faster than the state-of-the-art approaches (The fastest approach is 15 seconds per frame for full body tracking). We achieve this fast processing speed by analyzing human motion in a hierarchical way. Multiple persons are tracked robustly during large movement with severe inter-person occlusions and self occlusions.

In Chapter 2, we started with single person detection and tracking using 2D approaches. Single person tracking is a simpler problem compared with multiple people tracking and we achieved real-time performance. We demonstrated a spatial game application based on the proposed 2D approach. Next, this 2D approach was extended into two person tracking (Chapter 3). Two identical trackers were used to track each individual respectively. In order to deal with inter-person occlusions, the torso appearance was used as another feature to distinguish the front person and the back person. The experimental results showed that the front person was tracked accurately and robustly during occlusion. However, for the back (occluded) person, the estimated pose is just a copy of the one in the last frame before occlusion. If the occluded person changes his/her pose during occlusion, the estimated poses will not be correct. Due to the view limitation of single camera, we could not further recover the pose of the occluded person. Therefore, we explored multiple view approaches to solve this problem in Chapter 4. The previous used 2D model was replaced by a 3D model. Additionally, occlusion reasoning was taken into account to handle inter-person occlusions and self occlusions. The proposed approach is faster than the existing methods with comparable accuracy. In Chapter 5, the multiple people tracking results, i.e. the 3D joints positions, are used for interaction representation and recognition. The proposed approach provides an easy solution for unknown-time-duration interactions representation and recognition by incorporating the temporal and spatial structure of each interaction and it does not depend on the specific type of interactions.

Chapters 2-5 form a complete framework for vision-based human motion analysis, which allows us to understand video content from a low-level to a high-level (Figure 1.1). Each step in Figure 1.1 also can be studied separately given the required inputs, which is convenient from both theoretical and practical point of views.
6.1 Vision-based applications

Our proposed framework does not only contribute to academic research in vision-based human motion analysis, but also provides practical solutions to innovative human computer interactions. In the serious gaming industry, we expect more realistic virtual characters, more natural virtual worlds and more playful interactions in the coming decade. In the education sector, learning can be more fun if it is in a flexible and physical environment. In the case of language learning, pupils can explore language using their own poses and communicate with technologically enhanced interactive objects. This new educational format bridges formal and informal learning. In the healthcare sector, there is a strong desire to transfer parts of the care program, such as physical therapy, from hospitals and care centers to homes. Not only because this will reduce the healthcare costs; but also because patients can benefit from recovering in their own familiar environment with family and friends. Vision-based human motion analysis is expected to be one of the key elements to make this transfer possible. In the safety sector, human motion analysis is used in training safety procedures and crisis management. Nowadays, it is already possible to take part of the driving lessons in a simulator.

6.2 Future directions

In order to tackle challenges in the future, we expect more and more vision-based solutions for possible future applications.

6.2.1 Speedup of a tracking system

In our tracking system, we use multiple cameras (3 or 4 synchronized RGB cameras) to record moving people. Obviously, the more cameras are used, the more data we will have. The advantage of having more data is that we can extract richer features from these data, leading to more accurate tracking. On the other hand, it also takes more time to process the larger amount of data, which will limit the speed of the tracking system.

One possible solution to speed up the system is to use graphical processing units (GPUs). The computing power of GPUs is orders of magnitude greater than that of central processing units (CPUs) for certain operations. Nowadays, GPUs are considered as essential resources for scientists and developers to tackle their computational challenges. For computer vision and pattern recognition tasks, massive computation can be performed in parallel using multiple GPUs, leading to a faster process compared with a CPU implementation [119]. For instance, the HOG (Histograms of Oriented Gradients) based pedestrian detection algorithm implemented on a GPU is 8 times faster than on a CPU [120]. Meanwhile, OpenCV (Open Source Computer Vision), a library of programming functions for real time computer vision, released a GPU module in June, 2011 [121]. This module provides developers with a convenient computer vision framework that makes use of the GPU.

Another possibility to achieve faster tracking is to combine RGB cameras with other motion sensory equipment such as the Kinect [122-125] or RGB-D sensors [126] to capture
6.2. Future directions

body movements. The Kinect is developed by Microsoft for the Xbox video game console. It allows users to get rid of controllers and play games using body movements and poses. A Kinect device consists of a RGB camera and depth sensor, which captures 3D full-body motion. The Kinect is able to simultaneously track up to 6 persons in real time. However, since there is only one RGB camera integrated in the device, the Kinect has its limitation in multiple people occlusion handling. Besides, one Kinect device only can analyze two persons’ joint motion in parallel. To solve this limitation, Kinect can be synchronized with RGB cameras, or multiple Kinect devices can be combined together to increase the number of tracked person and achieve better occlusion handling.

6.2.2 Benchmark dataset

To the best of our knowledge, so far there is no benchmark dataset available for multiple people pose estimation with ground truth data. A first effort was made by the Utrecht University. They proposed a 3D dataset for multi-person motion analysis [128]. Due to the lack of benchmark datasets, the quantitative comparison of different approaches is currently difficult. Nowadays the evaluation of existing approaches is only done with own recorded video sequences and mainly relies on visual inspection of results. This is achieved by visually evaluating how the estimated pose matches image evidences [43].

The lack of benchmark data can be attributed to the difficulty of obtaining ground truth of multiple people 3D body pose. Even with a commercial motion capture (MoCap) system from ViconPeak [127], a lot of manually labeling and correction is required to align makers with corresponding body joints, specifically when multiple people are tracked simultaneously. For instance, markers might be lost during tracking and re-appear again, or the identity of makers may switch. Together this makes it difficult to get 3D ground truth data for multiple people tracking.

In 2011, a 3D dataset for multi-person motion analysis was proposed; the Utrecht Multi-Person Motion (UMPM) benchmark dataset [128]. This dataset provides multiple synchronized video sequences and motion capture data [129]. The scene includes multi-person motion and multi-person interaction. The dataset is meant for quantitatively evaluate multiple people tracking and pose estimation algorithms. Compared with HumanEva [43], the UMPM benchmark dataset has the advantage of capturing multi-person motion simultaneously. The disadvantage of the UMPM benchmark is that the videos are recorded with low resolution and the size of person is small, which makes tracking of a person’s limbs difficult. Although the UMPM dataset is not well-known and has its own limitations, it can be considered as a preliminary dataset for evaluating multiple people tracking algorithms.

6.2.3 Fully-automatic initialization

In Chapter 2 of this thesis, we proposed an approach for single person detection and tracking. This approach is able to automatically track a single person in 2D without manually setting any model parameters such as size or location in advance. The initialization is fully automatic due to the simplicity of the model. Further we extended this 2D approach to two persons tracking with possible inter-person occlusions (Chapter 3). A more complex 2D model is used, which includes the appearance of the tracked person. To build the appearance model, an initial pose is needed (Figure 3.3). As for the 3D tracking presented in Chapter 4,
the initialization step also includes using an initial pose. However, when this initial pose is not available in video data, manually labeling is required for initialization, which is a limitation of a tracking system. Therefore, one possible extension of our current tracking system is to realize fully automatic initialization, which might be achieved through automatic body parts detection [130-132].

For instance, before tracking starts or when tracking is lost, detection will take place. Different body parts, such as head, torso, upper arms and lower arms, can be detected and assembled together to form a whole body configuration (bottom-up approach). This body configuration will initialize the parameters needed in a model-based tracking system (top-down approach). In this way, bottom-up approaches can compensate shortcomings of top-down approaches. The combination of bottom-up and top-down approaches could be an effective way to initialize or re-initialize a multiple people tracking system.

6.2.4 Particle filter based tracking

Particle filter based approaches have been successfully used in object tracking. However, because it is a stochastic method and only past observations are taken into account, the tracking output is not very smooth. In [41], some smoothing filters are employed to improve particle-filtered estimations. However, the results show that existing smoothing methods are unable to provide much improvement, due to many parameters involved in body tracking. Therefore, reducing the search dimensionality of a particle filter might be crucial for improving tracking accuracy (Chapter 4). Another way to get more accurate tracking is to combine various image features, such as color, texture, and edge, and set the weighing factors of different features adaptively to the scene. This needs prior knowledge of each individual and the environment. Machine learning algorithms can be used to learn what are the most informative features to describe a body part.

With the investment of more resources, there is room for much more achievements. Vision-based applications hold the promise to change almost every aspect of the way we live, such as education, health-care, communication, training, entertainment, etc.
Bibliography


[59] http://gate.gameresearch.nl/

[61] http://en.wikipedia.org/wiki/Motion_capture#cite_note-1


http://www.pong-story.com/rhbaer.htm

http://www.ogre3d.org/

http://en.wikipedia.org/wiki/Internet_socket


Summary

In the last decade, computer vision has drawn more and more attention because of its potential applications in our daily lives, such as health care, education, safety, and training. However, the high complexity of many of the vision-based approaches hinders their practical applications, especially in applications where immediate feedback is required. This thesis focuses on developing fast and robust algorithms to track multiple people in a scene, which can be used in applications like interactive education, crisis analysis, serious games, or virtual reality.

We proposed a complete system for people tracking and interaction recognition. This system consists of three modules: detection and tracking persons from a single camera view, position estimation of body parts of each person from multiple camera views, and human pose and interaction recognition.

In the first module, we detect and track people in a single camera view. A simple and efficient approach is proposed to track a single person’s motion in real time. Based on this approach, we designed a pose-driven spatial game, which demonstrated a practical vision-based application. Further, we extended this single person tracking approach into multiple people tracking, by using color information to distinguish persons. Due to the simplicity of the features and the simplified model, the tracker is able to track two persons simultaneously in close real-time.

In the second module, we investigate multiple view approaches for multiple people tracking and pose estimation. The advantages of multiple view approaches over single view approaches are more camera views and better depth estimation. Multiple view approaches can lead to more accurate tracking compared with single view approaches, especially for multiple persons tracking. We proposed a novel approach that combines multiple views in such a way that we rely more on the features derived from clear views and less on those from occluded views. In this way, the proposed approach leads to accurate tracking results at much lower computational costs.

In the third module, we use the tracking results (positions of body parts) to represent and recognize human interactions. We investigate what are the most informative features to distinguish the interactions while keeping good recognition performance. The necessity of informative feature lies in practical applications. By using the informative features, even a linear classifier is shown to be sufficient for interaction recognition. This is quite attractive from a computational point of view.

Our proposed approaches do not only contribute to academic research in vision-based human motion analysis, but also provide practical solutions to innovative human computer interactions.
Samenvatting

In het afgelopen decennium heeft computer vision meer en meer aandacht gekregen vanwege het enorme aantal toepassingen in ons dagelijks leven, zoals gezondheidszorg, educatie en beveiliging. Veel van de op video gebaseerde toepassingen zijn lastig toe te passen vanwege de hoge mate van complexiteit, zeker als er een directe terugkoppeling wordt vereist. Dit proefschrift is gericht op het ontwikkelen van snelle en robuuste algoritmen voor het detecteren en volgen van meerdere personen in een video opname. Deze algoritmen kunnen gebruikt worden in toepassingen zoals interactieve educatie, crisis analyse, serious games en virtual reality.

Een compleet systeem voor het detecteren en van mensen en hun interacties is ontwikkeld. Dit systeem bestaat uit drie modules: detecteren en volgen van personen met een enkele camera opname, locatie schatting van ledematen van elk persoon met meerdere camera opnames en lichaamshouding en persoons interactie herkenning.

De eerste module detecteert en volgt personen vanuit een enkele camera opname. Een simpele en efficiënte aanpak wordt voorgesteld om de bewegingen van een enkele persoon te volgen in real time. Gebaseerd op deze aanpak is een pose-driven spatial game ontworpen, wat een praktische vision-based toepassing demonstreert. Daarnaast is deze aanpak uitgebreid om meerdere personen te kunnen onderscheiden door middel van kleur informatie en individueel te kunnen volgen. Vanwege de lage mate van complexiteit van de kenmerken en het versimpelde modelleren is het mogelijk om twee personen simultaan te volgen in bijna real-time.

In de tweede module wordt een aanpak voor het detecteren en volgen van meerdere personen met meerdere camera’s onderzocht. Het voordeel van een aanpak met meerdere camera’s ten opzichte van een aanpak met een enkele camera is een betere diepe schatting. Een aanpak met meerdere camera’s leidt ook tot het accuraat volgen van personen ten opzichte van een aanpak met een enkele camera. Een nieuwe benadering die meerdere camera opnames combineert waarbij de nadruk meer ligt op goed zichtbare kenmerken en minder op overlappende kenmerken wordt geïntroduceerd. De geïntroduceerde aanpak leidt op deze manier tot betere volg resultaten bij een lagere computatiesnelheid.

De derde module gebruikt de volg resultaten om de persoons interacties (posities van de ledenmaten) te herkennen en representeren. Er wordt onderzocht wat de meest informatieve kenmerken zijn om onderscheid te maken tussen interacties terwijl de herkenningsprestaties goed blijven. De noodzaak van informatieve kenmerken komt vanuit praktische toepassingen. Door de informatieve kenmerken kan zelfs een linear classifier gebruikt worden voor persoons interacties. Dit is erg voordelig vanuit het oogpunt van computatiesnelheid.
De geïntroduceerde aanpak is niet alleen een contributie voor wetenschappelijk onderzoek naar vision-based human motion analysis, maar is ook een praktische oplossing voor innovatieve mens-machine interacties.
Acknowledgements

During my PhD studies, I have received a lot of help and support from many people. They gave their contributions to this thesis either directly or indirectly. Here I want to take this opportunity to give my gratitude to them.

First of all, I would like to thank my supervisor Emile Hendriks for his great supervision. It is very pleasant and enjoyable to work with you. As a teacher, you know exactly how to lead a student. As a colleague, you leave the research initiative to me. As a friend, you show us your hospitality for our every visit. All the progress and improvement that I had in the last years are inseparable from your contributions. Inside the TU, we had inspiring discussions on research. Outside the TU, you showed us the totally different countryside life. We enjoyed our every visit to you and Ans.

Secondly, I would like to give my gratitude to my promotor Marcel Reinders. I appreciate the feedback and comments that you gave on my thesis. I also want to thank you for arranging the committee and providing all the needed documents for my graduation.

This thesis involved many people’s efforts and collaboration. I would like to thank Parvel and Carmen for their contribution and helpful discussions. With the pattern recognition toolbox developed in perClass, we could build a nice life demo to show our research in an easily understandable way.

In the last four years, I had a close cooperation with the Multimedia and Geometry group in Utrecht University. I would like to thank: Berend, Xinghan, Nico, Pascal, Remco, Wolfgang, and Robby for the frequent discussions. Thank you, Berend, Xinghan, and Nico for helping me with the recording of the dataset used in this thesis.

I had a pleasant time in the Intelligent System Group. The staff members in the group are always willing to give me their support and help me out. I would like to thank you, Jan Biemond, Bob Duin, Alan, Boudewijn, Marco, Dick, Jeroen, Richard Heusdens, Richard Hendriks, Guoqiang, Zekeriya, and Martha for your help and support. David, thank you for always being open for questions and enthusiastic for discussions. Laurens, thanks for giving us nice lectures and providing me all kinds of useful information. Thank you: Robbert and Saskia for your excellent support.

I would like to thank my colleagues who I shared my daily lives with. My gratitude goes out to Wan-Rui, Lu, Yan, Viet, Alessandro, Gorkem, Veronika, Ahmed, Vikas, Yuanhao. Thank you all for the formal and informal talks. Lu, thank you very much for your company in the office. You are always happy and cheerful. Since you joined the office, my working days always started with funny stories and jokes. Alessandro, thank you for helping me with Dutch and thanks for the nice home-made Italian food. I would like to thank Carsten, Cees, Jorge, Cynthia, Christoph, Michael, Peng, Raynor, Steven, Wen, Xinchao, Yuan, and Yue for the company during lunch breaks and all the social activities. Thank you also, Sepideh, Shriprakash, Thies, Bastiaan, Jan and Jurgen for all the PRB borels.
I would like to thank all my family members and friends, both in China and in the Netherlands, for their continuous care and precious support. I am so lucky to have you in my life. Great appreciation goes to my parents. Thank you for your unconditional love. Special thanks go to my husband Michiel and my daughter Jane for all the wonderful moments with you.
Curriculum Vitae

Feifei Huo was born in Luoyang, China, on October 27, 1979. She obtained her Bachelor and first Master degree in Electrical Engineering from Xidian University, China, in 2003 and 2006 respectively. In August 2006, she came to the Netherlands for her second Master study. Her graduation project was on people detection and tracking in an indoor environment, under the supervision of Emile Hendriks and Stijn Oomes, at Delft University of Technology.

In March 2008, she started her PhD studies, at the Intelligent System Group, under the supervision of Emile Hendriks and Marcel Reinders. She conducted research on multiple people tracking, modeling, and interaction recognition. This research was part of the GATE (Game Research for Training and Entertainment) project. During her PhD, she had a close cooperation with the Multimedia and Geometry group, at Utrecht University.

Since November 2012, she has been working in Almende B.V. as a scientific developer. She is working on the SEAM4US project. Her work focuses on passenger occupancy and flow estimation from surveillance videos.

List of Publications
