UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

Human machine interactivity using vision-based posture analysis at multiple levels

Permalink

https://escholarship.org/uc/item/4tr1g5q5

Authors

Tran, Cuong
Tran, Cuong

Publication Date

2012

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Human Machine Interactivity using Vision-based Posture Analysis at Multiple Levels

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Computer Science

by

Cuong Tran

Committee in charge:

Professor Mohan M. Trivedi, Chair
Professor Serge J. Belongie
Professor Garrison W. Cottrell
Professor David J. Kriegman
Professor Bhaskar D. Rao

2012

Copyright

Cuong Tran, 2012
All rights reserved.

The dissertation of Cuong Tran is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2012

DEDICATION

To my beloved parents.

EPIGRAPH

Everything should be made as simple as possible,
but not simpler.
—Albert Einstein

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank my family, mentors, colleagues, and friends whose help, cooperation, and support have made this dissertation possible.

First and foremost, I would like to thank my advisor, Professor Mohan Trivedi, for his wonderful guidance, mentorship, inspiration, enthusiasm, patience, and determination. He has provided me with much-needed support and confidence to enable me to complete my studies. I would also like to thank my committee members, Professors Serge Belongie, Garrison Cottrell, David Kriegman, and Bhaskar Rao, for their time, expertise, and constructive advice, which have enriched my research.

The advice, inspiration, knowledge, as well as assistance in conducting experiments and holding deep relevant discussions of all my colleagues and predecessors from the Computer Vision and Robotics Research Laboratory and beyond will never be forgotten. I would especially like to acknowledge the collaborations and efforts of Dr. Shinko Cheng, Dr. Anup Doshi, Dr. Brendan Morris, Ashish Tawari, Sujitha Martin, Michael Holte, Matt Wilder, Professor Mike Mozer, and Professor Thomas Moeslund.

I am thankful for the support of the Vietnam Education Foundation Fellowship from 2006 to 2008. Parts of the dissertation are sponsored by the UC Discovery Program, Volkswagen, and the National Science Foundation, to all of whom I am indebted.

I would like to thank everyone in my extended family for their caring, patience, and understanding. Most importantly, thanks to my dearest parents and brother, for their undying, unquestioning love and support.

Publication acknowledgments: Chapter 2 is based on material that is published in the IEEE Journal of Selected Topics in Signal Processing (2012) by Michael B. Holte, Cuong Tran, Mohan M. Trivedi, and Thomas B. Moeslund. Chapter 3 is based on material that is published in the International Conference on Pattern Recognition (2008) and IEEE Transactions on Industrial Informatics (2012) both by Cuong Tran and Mohan M. Trivedi. Chapter 4 is based on material that is published in the IEEE Transactions on Industrial Informatics (2012), IEEE International Conference on Vehicular Electronics and Safety (2009) both by Cuong Tran and Mohan M. Trivedi, and Computer Vision and Image Understanding — Elsevier (2012) by Cuong Tran, Anup Doshi, and Mohan M. Trivedi. Chapter 5 is based on material that is published in the IEEE Intelligent Vehicles Symposium (2011) and IEEE International Conference on Intelligent Transportation Systems (2012) both by Cuong Tran, Anup Doshi, and Mohan M. Trivedi. The "Sequential dependencies in driving" section (Section 5.3) is also based on material published in the Cognitive Science (2012) by Anup Doshi, Cuong Tran, Matthew H. Wilder, Michael C. Mozer, and Mohan M. Trivedi.

VITA

2004	B. S. in Computer Science, Hanoi University of Technology, Hanoi, Vietnam
2006-2012	Graduate Student Researcher, University of California, San Diego
2008	M. S. in Computer Science, University of California, San Diego
2012	Ph. D. in Computer Science, University of California, San Diego

PUBLICATIONS

Cuong Tran and Mohan M. Trivedi, “3D Posture and Gesture Recognition for Interactivity in Smart Space”, *IEEE Transactions on Industrial Informatics, vol. 8, no. 1, pp. 178-187*, 2012.

Cuong Tran, Anup Doshi, and Mohan M. Trivedi, “Modeling and Prediction of Driver Behavior by Foot Gesture Analysis”, *Computer Vision and Image Understanding, vol. 116, no. 3, pp. 435-445*, 2012.

Michael B. Holte, Cuong Tran, Mohan M. Trivedi, and Thomas B. Moeslund, “Human Pose Estimation and Activity Recognition From Multi-View Videos: Comparative Explorations of Recent Developments”, *IEEE Journal of Selected Topics in Signal Processing, vol. 6, no. 5, pp. 538-552*, 2012.

Anup Doshi, Cuong Tran, Matt H. Wider, Michael C. Mozer, Mohan M. Trivedi, “Sequential Dependencies in Driving”, *Cognitive Science, vol. 36, no. 5, pp. 948-963*, 2012.

Brendan Morris, Cuong Tran, George Scora, Matthew Barth, and Mohan M. Trivedi, “Real-Time Visual Traffic Measurement and Visualization Tool”, *Transactions on Intelligent Transportation Systems*, to appear 2012.

Cuong Tran, Anup Doshi, and Mohan M. Trivedi, “Investigating Pedal Errors and Multi-modal Effects: Novel Driving Testbeds and Experimental Analysis”, *IEEE Conference on Intelligent Transportation Systems*, 2012.

Cuong Tran and Mohan M. Trivedi, “Vision for Driver Assistance: Looking at People in a Vehicle”, in T. Moeslund, A. Hilton, V. Krger, and L. Sigal (Eds.), *Visual Analysis of Humans: Looking at People (pp. 597-614), Springer*, 2011.

Cuong Tran, Anup Doshi, and Mohan M. Trivedi, “Pedal Errors Prediction by Driver Foot Gesture Analysis: A Vision-based Inquiry”, *IEEE Intelligent Vehicle Symposium*, June 2011.

Cuong Tran and Mohan M. Trivedi, “Towards a Vision-based System Exploring 3D Driver Posture Dynamics for Driver Assistance: Issues and Possibilities”, *IEEE Intelligent Vehicle Symposium*, June 2010.

Cuong Tran and Mohan M. Trivedi, “Driver Assistance for Keeping Hands on the Wheel and Eyes on the Road”, *IEEE International Conference on Vehicular Electronics and Safety*, November 2009.

Cuong Tran and Mohan M. Trivedi, “Introducing XMOB: Extremity Movement Observation Framework for Upper Body Pose Tracking in 3D”, *IEEE International Symposium on Multimedia*, December 2009.

Cuong Tran and Mohan M. Trivedi, “Hand Modeling and Tracking from Voxel Data: An Integrated Framework with Automatic Initialization”, *IEEE International Conference on Pattern Recognition*, December 2008.

Cuong Tran and Mohan M. Trivedi, “Human Body Modeling and Tracking Using Volumetric Representation: Selected Recent Studies and Possibilities for Extensions”, *AMMCSS workshop, IEEE Int’l. Conf. on Distributed Smart Cameras*, September 2008.

ABSTRACT OF THE DISSERTATION

Human Machine Interactivity using Vision-based Posture Analysis at Multiple Levels

by

Cuong Tran

Doctor of Philosophy in Computer Science

University of California, San Diego, 2012

Professor Mohan M. Trivedi, Chair

This dissertation focuses on vision-based articulated body pose tracking and human activity analysis for interactive applications, e.g., intelligent driver assistance systems, gesture-based interactive games, and smart rooms. Although there has been a considerable amount of related research effort, developing real-time, robust, and efficient vision-based systems for real-world interactive applications is still an open and important research area. Since human activities in the real-world may happen at different levels of detail (e.g., full body, upper body, hands, head, and feet), it is desirable to have systems that look at humans at multiple levels for better understanding of their activities.

In addition to common computer vision issues such as occlusion, background clutter, variable lighting condition, and human appearance, there are other research issues that need to be addressed, with the objective of analyzing human posture at multiple levels for interactivity. This dissertation discusses those issues and proposes several relevant frameworks and approaches for human posture and activity analysis at different levels of detail.

First, in order to achieve real-time performance and robustness required for interactive applications, the trade-offs in developing generic versus application-specific approaches for efficiency should be considered. Focusing on applications like driver assistance systems and smart meeting rooms, we develop the very first system, as far as we are concerned, that does both real-time upper body pose tracking in 3-D by observing extremity movements and then gesture recognition using pose tracking output.

Second, it is more feasible to a deploy multilevel posture analysis system if we have efficient algorithms which may apply to different body levels. In that regard, we develop an integrated framework with automatic initialization for body and hand modeling and tracking from 3-D voxel data. We also develop an optical flow-based framework for driver foot and head behavior modeling and prediction which shows potential in mitigating the incidents of pedal misapplication in the real world.

Third, we develop a driver assistance system that combines information from driver head and hand activities for distraction monitoring.

Lastly, we present our development of multimodal driving experiments and analysis based on the ability to track driver activities at different levels which provides insight into the effect of audio and visual cues on driver behavior.

Chapter 1

Introduction

Human machine interactivity for intelligent systems has emerged as an important and interdisciplinary area. The goal is to develop better and more naturalistic interfaces between intelligent systems and humans in order to make computer technology more usable by people. In the communication between humans and machines, although humans can easily perceive different types of input such as text, audio, or visual, the other direction is typically more limited. Traditionally, machines only understand inputs like computer mouse or keyboard. In more recent years, there has been a significant growth in research and technology development in order to improve the ability of computer systems in understanding human activities, intentions, and affective states from a more naturalistic medium such as audio, visual, and touch input. This is a broad research area which involves various research studies of its different aspects including studies on tactile interface [9], audio analysis for affect recognition [168] and speech recognition [38], vision analysis for gesture and activity recognition [32, 52, 92], as well as the fusion of multimodal information for human machine interface and intelligent systems [62, 124]. Some of those technologies have also been commercialized such as the Microsoft Kinect Xbox for human motion sensing from a structured light infrared projector and sensor system or the iPhone Siri application for natural language user interaction. However, this is obviously not a solved area yet and more research efforts are still needed. There is a lot of room for research and technology development in terms of various application areas such as intelligent driver assistance, assisted living, smart environments, or surveillance (more details will be discussed in Chapter 2) as well as the improvement of accuracy, robustness, and real-time performance in challenging real-world conditions which is required for those applications.

In this dissertation, we focus on developing vision-based intelligent systems which can observe and interpret human activity for interactive applications such as intelligent driver assistance systems, gesture-based interactive games, and smart rooms. To elaborate, two main points in the focus are intelligent systems using vision input (an important channel of information) versus using audio or tactile input and intelligent systems for interactive applications (with online user interactions) versus offline human activity analysis. There has been a considerable amount of related research studies with goal of developing marker-less, nonintrusive vision-based systems for human activity analysis. However developing real-time, robust, and efficient vision-based systems for human machine interactivity is still an open and challenging research area. Human activities may happen at different levels of detail such as full body, upper body, hands, head, and feet. Related research studies typically focus only on a single body level, from gesture analysis of full-body level (e.g., walking, running, jumping, bending gestures [54, 159, 167]) to upper-body level (e.g., waving, punching, and driver sitting pose [16]), to more detailed levels of hand and facial gestures (e.g., hand gesture interacting with objects [106, 155], sign language [40, 69, 154], face pose tracking [55], and facial action coding [7]). Nevertheless, natural human activities generally involve multiple body levels. For example, there could be different gestures of upper body, head, hand, and foot while people are driving or talking. Therefore, it is desirable to have systems that look at humans at multiple levels to better understand their activities. With that motivation, this dissertation proposes several relevant frameworks and approaches for human posture and activity analysis at different levels of detail. Along with those works, we will also discuss how we dealt with some issues related to the emphasis of interactive applications, such as utilizing inputs from user interaction to help the system, or the trade-offs in developing application-specific versus generic approaches for efficiency (e.g., real-time performance) and robustness which are particularly important for interactivity.

1.1 Problem Statement and Challenges

The ultimate goal is to build vision-based, interactive systems that look at people at different levels of details to learn, to perceive human articulated body pose and activity, as well as to provide assistance to people when needed. Below is a clarification of some terms and their meaning as being used in the scope of this dissertation.

- Articulated body pose: Refers to the kinematic model of human body in which a pose is determined by the position and orientation of each body part (e.g., torso, upper arms, lower arms, etc.). Typically, the dimensions of each body part are considered fixed.

- Activity: Refers to “a process that an organism carries on or participates in by virtue of being alive” *(Merriam-Webster).* This is used as a general term which may imply (a set of) gestures and/or behaviors.

  Gesture: Is “the use of motions of the limbs or body as a means of expression” *(Merriam-Webster)* (action)

  Behavior: Is “the response of an individual, group, or species to its environment” *(Merriam-Webster)* (reaction)

- Multilevel pose and activity analysis: Refers to the multiple levels of details such as full body, upper body, head, hands, and feet.

The automatic perception of human posture and activity from vision input is a challenging task due to the occlusion issue, background clutter, variable lighting condition and human appearance, variance in the way people perform gestures and activities. In addition to those common computer vision challenges, there are some other research issues related to the objective of multiple level analysis and interactivity emphasis including:

- How do we efficiently track and utilize information from multiple levels of detail for better human activity analysis? We may want to develop some efficient analysis frameworks which can be easily adapted to different body levels instead of developing totally separate systems for each level.
- How do we deal with the typical trade-offs between achieving detailed information of human gesture at different levels and the efficiency (e.g., real-time performance) as well as robustness which are particularly important for interactive applications? Depending on specific applications, we may need to select different levels of detail which provide useful information about the concerned activities and then develop algorithms focusing on those levels.
- How do we utilize the interaction with user? In interactive applications, the user and the system work in collaboration. Therefore, the system might also expect some supporting feedback from the user which could help to ease the difficulties and improve system performance.

1.2 Contributions and Outline

With the objective of tracking and analyzing human posture and activity at multiple levels of detail for interactivity, this dissertation proposes several relevant frameworks and approaches which address to some extent the challenges mentioned above. The novel findings and contributions of this dissertation include the following:

- Develop an integrated framework with automatic initialization for human body and hand modeling and tracking from voxel data.
- Introduce XMOB (Extremity Movement Observation) framework for real-time upper body pose tracking in 3-D.
- Develop the very first system, as far as we are concerned, that does both real-time upper body pose tracking in 3-D using XMOB and then gesture recognition based on pose tracking outputs.

- Develop a driver assistance system for keeping hands on the wheel and eyes on the road.
- Develop the very first vision-based system, as far as we are concerned, for driver foot behavior modeling and prediction (towards mitigating pedal errors).
- For the first time, in complex environments, we provide some quantitative analysis of the effect of audio-visual cues on driver behavior such as the reaction time, movement time, and number of pedal errors.

A significant portion of this work included the development and engineering of several testbeds and databases:

- Develop LISA-P real-world driving testbed with multimodal inputs and displays.
- Develop LISA-S driving simulator testbed with multimodal inputs and displays for basic, controlled laboratory studies of human behavior, interactivity, and cognition.
- Develop joint audio-visual experiments and databases.

The rest of this dissertation is organized as follows: Chapter 2 reviews the literature of related research studies in human pose estimation and human activity analysis focusing on multiview approaches. In Chapter 3, we present our works in human pose modeling and tracking at different levels of detail including a framework with automatic initialization applied for both body and hand, an approach for real-time upper body pose tracking in 3-D based on extremity movement observations, as well as combining pose tracking outputs from different levels of detail. In Chapter 4, we discuss our works on vision-based human activity analysis for interactivity using information from different body levels (how we deal with some research issues mentioned earlier) including upper body activity analysis based on pose tracking output, combining head and hand activity for distraction monitoring, and a framework for driver foot and head behavior modeling and prediction. In Chapter 5, we then describes the development of several multimodal testbeds and driving experiments as well as our analysis which provided some insights into the effect of different cue modalities on driver behavior. Finally, Chapter 6 is the concluding remarks and discussion.

Chapter 2

Review of Recent Developments in Human Pose Estimation and Activity Recognition

This chapter provides a brief overview of related research studies in human pose estimation and activity recognition. We begin with a discussion of application domain (covering advanced Human-Computer Interaction (HCI), assisted living, gesture-based interactive games, intelligent driver assistance systems, movies, 3-D TV and animation, physical therapy, autonomous mental development, smart environments, sport motion analysis, video surveillance, and video annotation) as well as the associated requirements on human pose estimation and activity recognition. Next, we will present a review and comparative study of selected approaches. Since human pose estimation and activity recognition is a broad area, we narrow down our discussion into recent developments in 3-D human pose estimation and activity recognition from multiview video inputs.

2.1 Introduction

In recent years a wide range of applications using 3-D human body modeling, pose estimation, and activity recognition has been introduced. Among those, several key applications are illustrated in Figure 2.1 including the following

- Advanced Human-Computer Interaction (HCI): Beyond the traditional medium like computer mouse and keyboard, it is desirable to develop better, more natural interfaces between intelligent systems and human in which understanding visual human gesture is an important channel. A few examples are using hand movement to control the presentation slides

[74] or recognizing manufacturing steps to help workers to learn and improve their skills [114].

- Assisted living: Pose estimation and activity recognition can also be applied in assisting handicapped people, elderly people, as well as general users. For example, a system to detect when a person falls [122] or a robot controlled by blinking [4].

- Gesture-based interactive games: These games allow the player to use nonintrusive body movement for interaction. For example, an Interactive Balloon Game [140] or the well-known Microsoft Kinect Xbox [125].

- Intelligent driver assistance systems: Looking at the driver is a key part which is required in a holistic approach for intelligent driver assistance systems [146]. Examples of driver assistance systems using posture and behavior analysis are: monitoring driver awareness based on head pose tracking [97], combining driver head pose and hands tracking for distraction alert [143], modeling driver foot behavior to mitigate pedal misapplications [138], developing a smart airbag system based on sitting posture analysis [148], or predicting driver turn intent [16].

- Movies, 3-D TV and animation: Human motion capture was also applied extensively in movies, 3-D TV and animation. For example, in the *Avatar* movie or in a digital dance lesson [41].

- Physical therapy: Modern biomechanics and physical therapy applications require the accurate capture of normal and pathological human movement without the artifacts of intrusive marker-based motion capture systems. Therefore, marker-less posture estimation and gesture analysis approaches were also developed to be applied in this area [117, 93].

- Smart environments: These environments have the ability to extract and maintain an awareness of a wide range of events and human activities occurring in these spaces [150]. For example, monitoring the focus of attention and interaction of participants in a meeting room [95, 158].

- Sport motion analysis: Several sports like golf, ballet, or skating require accurate body posture and movement. Therefore, posture estimation and gesture analysis could be applied to this area for analyzing performance and training.

- Video surveillance: Video surveillance is used in many places such as critical infrastructure, public transportation, office buildings, parking lots, and homes. However, the amount of surveillance data exceeds the abilities of human security guards to adequately monitor the scene. Therefore, approaches for automatic video surveillance including outdoor human activity analysis, e.g., [105], will be needed.

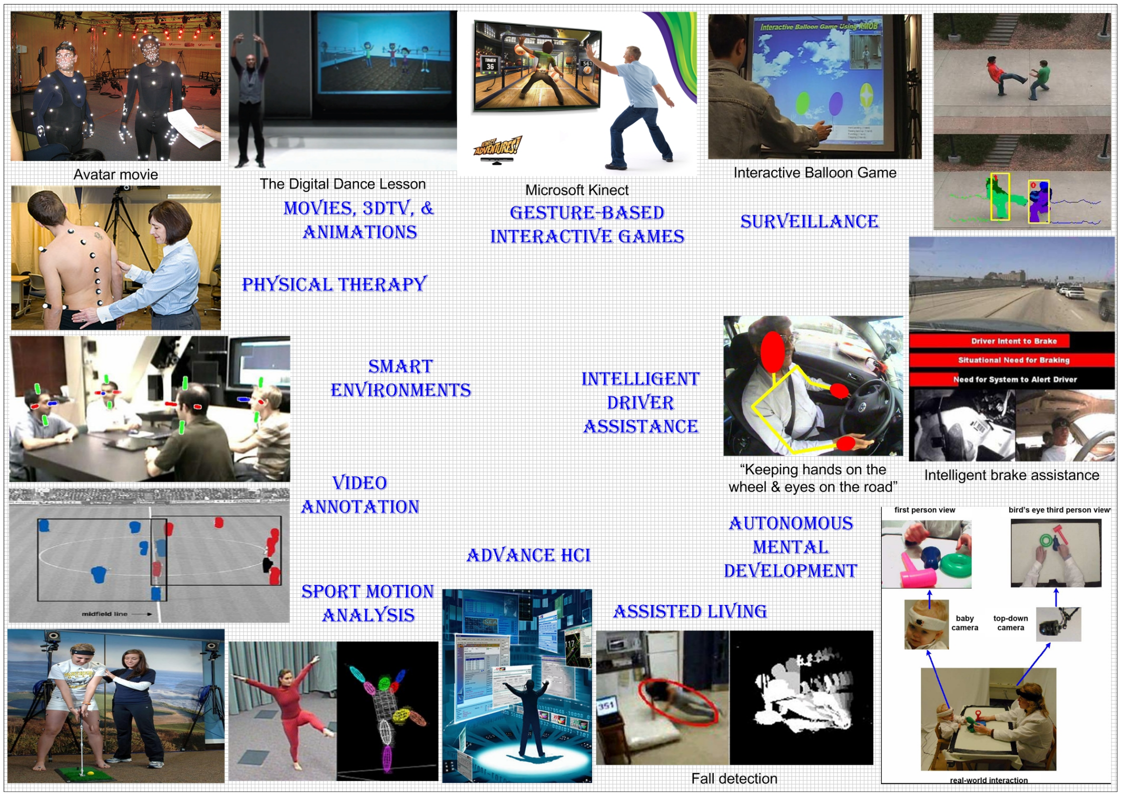


Figure 2.1: The application domain of human body modeling and motion analysis.

- Video annotation: With the development of hardware technology, a very large amount of video data can be easily saved. Among those, there are lots of human-related videos such as surveillance videos, sports videos, and movies. Instead of manually scanning through those large video databases to get the needed information, human motion analysis can be used to annotate those videos, for example approach to annotate the video of a soccer game [6].

Many approaches have been proposed to comply with the requirements of these applications based on different kinds of sensor systems for data acquisition: marker-based systems, laser-range scanners, structured light, Time-of-Flight (ToF) cameras, the Kinect sensor, and multicamera systems. Table 2.1 gives an overview of the application domain of human body modeling and motion analysis and the associated requirements. As shown in the table, the requirements vary significantly depending on the desired application. This results in the need of approaches, which can operate on different abstraction levels, in uncontrolled environments, with high precision, in critical real time and for large database search.

A number of surveys have been published during the last decade that review approaches for human motion capture, body modeling, pose estimation, and activity recognition in more general [63, 91, 92, 111, 112, 163, 164]. This chapter differs from those surveys in the sense that it focuses exclusively on recent work on multiview human body modeling, pose estimation, and action recognition, both based on 2-D multiview data and reconstructed 3-D data, acquired with standard cameras. Multiview camera systems have an advantage because they enable full 3-D reconstruction of the human body and to some extent handles self-occlusion. In contrast, single 3-D imaging devices, such as ToF sensors and Kinect, will only acquire 3-D surface structure visible from that single viewpoint. We give a more detailed description and comparison of some prominent and diverse 3-D pose estimation techniques, which properly represent the contributions to this field. Additionally, we present a quantitative comparison of several promising multiview human action recognition approaches using two publicly available datasets: the INRIA Xmas Motion Acquisition Sequences (IXMAS) Multiview Human Action Dataset [161] and the i3-DPost Multiview Human Action and Interaction Dataset [42].

2.1.1 Human Pose Estimation

Vision-based pose estimation and tracking of the articulated human body is the problem of estimating the kinematic parameters of the body model (such as joints position and joints angle) from a static image or a video sequence. Typically, the shape and dimension of body parts are assumed fixed, and the interdependence between body parts are only the kinematic constraints at body joints. Related research studies in this area include body pose estimation, hand pose estimation, and head pose estimation. Among those, the most extensive subfield is body pose estimation, which refers to the articulated body model normally with torso, head, and 4 limbs but without details of hand, foot, or facial variation. Several important applications explicitly required detailed 3-D posture information including movies and 3-D animation, sport motion analysis, physical therapy, as well as some application in advanced HCI or smart environments (e.g., robot controls or applications using pointing gesture). Moreover, the output 3-D pose information is also a rich and view-invariant representation for action recognition [112]. Developing an efficient and robust body pose estimation system, however, is a challenging task. One major reason is the very high dimensionality of the pose configuration space, e.g., in [17], 19 DOF (Degree Of Freedom) are used for the body model and 27 DOF are used for the hand model. As concluded in [126], although human tracking is considered mostly solved in constrained situation that has a large number of calibrated camera (> 10), people wear tight clothes, and the environment is static, there are still remaining key challenges including tracking with fewer cameras (< 4), dealing with complex environments, variations in object appearance (e.g., general clothes, hair, etc.), automatically adapting to different body shapes, and automatically recovering from failure.

Some surveys of several techniques for human body pose modeling and tracking can be found in [91, 92, 111, 164], each with different focus and taxonomy. Werghi [164] provided a general overview of both 3-D human body scanner technologies and approaches dealing with

Table 2.1: Different applications and their requirements to 3-D human body modeling, pose estimation, and activity recognition approaches.

Application	Abstraction level	Data acquisition	Precision	Time critical	Human body model
Advanced HCI	Full body, head, arms, hands, etc.	Mostly controlled	Medium-high	Real time	Mostly model-based
Gesture-based interactive games	Full body, head, arms, hands, etc.	Mostly controlled	Medium-high	Real time	Mostly model-based
Movies and 3-D animation	Full body, head, arms, hands, etc.	Controlled and uncontrolled	High	No	Model-based
Smart environments	Full body, head, arms, hands, etc.	Controlled and uncontrolled	Medium	Real time	Model-based and model-free
Video surveillance	Mostly full body	Mostly uncontrolled	Low-medium	Real time (mostly)	Mostly model-free
Intelligent driver assistance	Full body, head, arms, hands etc.	Controlled and uncontrolled	High	Critical real time	Model-based and model-free
Video annotation	Full body, head, arms, hands, etc.	Uncontrolled	Low-medium	No but desirable for large databases	Mostly model-free
Sport motion analysis	Full body, head, arms, hands, etc.	Controlled	High	No	Model-based
Physical therapy	Full body, head, arms, hands, etc.	Controlled	High	No	Model-based
Assisted living	Mostly full body	Controlled and uncontrolled	Medium	Real time	Model-based and model-free

such scanned data, which focus on one or more of the following topics: body landmark detection, segmentation of body scanned data, body modeling, and body tracking. In Poppe's survey on pose estimation techniques [111], he mentioned the division into 2-D approaches and 3-D approaches (depending on the goal to achieve 2-D pose or 3-D pose representation) and the division into model-based approaches and model-free approaches (depending on whether a priori kinematic body model is employed). Moeslund et al. [91] split the pose estimation process into initialization, tracking, pose estimation, and recognition. In [92], they also provided an updated review of advances in human motion capture for the period from 2000 to 2006. We see that it is not easy to have a unified taxonomy for the broad area of human body modeling and tracking. Quite similar to [91], we categorize related research studies based on the common components in a generic body pose estimation system. As shown in Figure 2.3, we first need a component to extract useful features from the input vision data, and then a procedure to infer body pose from extracted features. In this review and comparative study, we focus on representative model-based approaches using multiview video input and aim to extract a 3-D posture. In comparison to monocular view, multiview data can help to reduce the self-occlusion issue and provide more information to make the pose estimation task easier as well as to improve the accuracy. The underlying kinematic body model in model-based approaches can help to improve the accuracy and robustness, although it also raises the issue of model initialization and reinitialization.

2.1.2 Human Action Recognition

While 2-D human action recognition has received high interest during the last decade, 3-D human action recognition is still a less explored field. Relatively few authors have so far reported work on 3-D human action recognition [63, 92, 112, 163]. Human actions are performed in real 3-D environments. However, traditional cameras only capture the 2-D projection of the scene. Vision-based analysis of 2-D activities carried out in the image plane will therefore only be a projection of the actual actions. As a result, the projection of the actions will depend on the viewpoint and not contain full information about the performed activities. To overcome this shortcoming, the use of 3-D representations of reconstructed 3-D data has been introduced through the use of two or more cameras [1, 42, 60, 127, 161]. In this way, the surface structure or a 3-D volume of the person can be reconstructed, for example by Shape-From-Silhouette (SFS) techniques [133], and thereby a more descriptive representation for action recognition can be established.

The use of 3-D data allows for efficient analysis of 3-D human activities. However, we are still faced with the problem of determining the orientation of the subject in the 3-D space. Therefore, approaches have been proposed without this assumption by introducing view-invariant or view-independent representations. Another strategy which has been explored is the application of multiple views of a scene to improve recognition by extracting features from different 2-D image views or to achieve view invariance.

The ultimate goal is to be able to perform reliable action recognition applicable for, for example video annotation, advanced human computer interaction, video surveillance, driver assistance, automatic activity analysis, and behavior understanding. We contribute to this field by providing a review and comparative study of recent research on 2-D and 3-D human action recognition for multiview camera systems (see Table 2.3), to give people who are interested in the field an easy overview of the proposed approaches and an idea of the performance and direction of the research.

Methods for 3-D human action recognition can either be model-free or model-based. Mostly a model-free strategy is applied, which has the advantage of using a wide range of image, static shape/pose, and motion or statistical features, and it does not depend on a predefined human body model. However, the approaches usually do not capture any information about the 3-D human body pose, joint positions, etc. This limits its usability to a specific set of applications, where the exact pose and joint configuration of the body parts are not explicitly required (see Fig 2.1 and Table 2.1). Whereas, the model-based methods, which requires a human body model and are usually applied in conjunction with human body modeling and pose estimation, allows for description of the exact pose of the respective body parts. This opens up for another set of applications.

The remainder of the section is organized as follows. Section 2.2 is a review of selected recent model-based methods for human body pose estimation using multiview data. Section 2.3 gives a review of 2-D and 3-D approaches for human action recognition, followed up by a description of multiview dataset and a quantitative and qualitative comparison of promising methods. Finally in Section 2.4, we present a discussion and directions of future work.

2.2 Multiview 3-D Human Pose Estimation

As discussed earlier in Section 2.1.1, we will focus only on model-based approaches using multiview video input and aim to extract a real 3-D posture. Figure 2.4 shows common steps in model-based approaches for human pose estimation using multiview input including: camera calibration/data capture, voxel reconstruction, initialization/segmentation (segment voxel data into different body parts), modeling/estimation (estimating pose using current frame only), and tracking (use temporal information from previous frames in estimating body pose in current frame). In each step, different methods may have different choices of approaches. There are methods that use 3-D features (e.g., voxel data) reconstructed from multiple views while others may still use 2-D features (e.g., silhouette, contour) extracted from each view. They may have manual or automatic initialization step. Some methods may not have a tracking step. Some methods are for a generic purpose while others are application specific for efficiency. Table 2.2 is a summary of selected representative model-based methods for human body pose estimation using

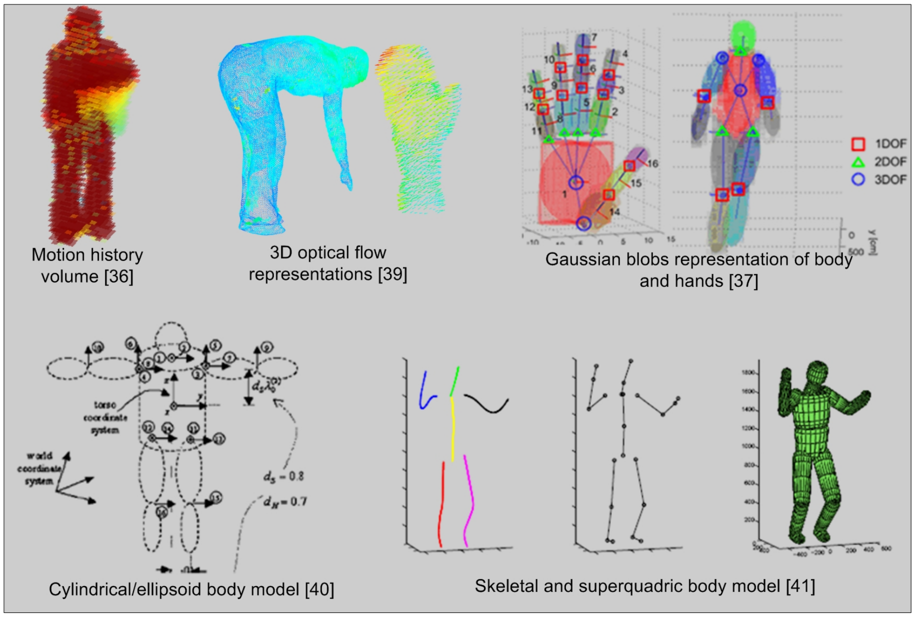

Motion history volume [36]
3D optical flow representations [39]
Gaussian blobs representation of body and hands [37]
1DOF
2DOF
3DOF
Cylindrical/ellipsoid body model [40]
Skeletal and superquadric body model [41]

Figure 2.2: Prominent 3-D human body model and human motion representations [17, 51, 89, 134, 161].

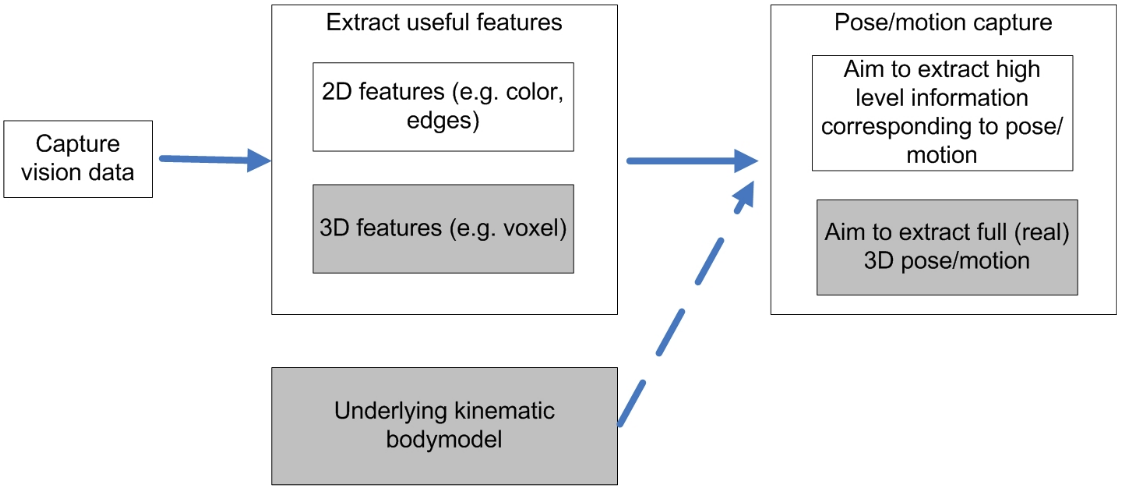

Capture vision data
Extract useful features
2D features (e.g. color, edges)
3D features (e.g. voxel)
Pose/motion capture
Aim to extract high level information corresponding to pose/ motion
Aim to extract full (real) 3D pose/motion
Underlying kinematic bodymodel

Figure 2.3: Block diagram of a generic human body pose estimation system. The dashed line means that the underlying kinematic model can be used or not. The gray boxes show the focus of this section, which are model-based methods that use voxel data and aim to extract a full 3-D posture.

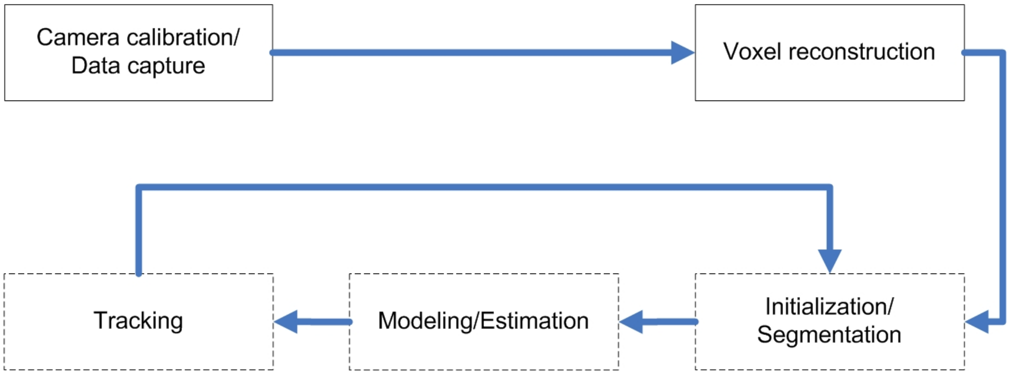

Camera calibration/ Data capture
Voxel reconstruction
Tracking
Modeling/Estimation
Initialization/ Segmentation

Figure 2.4: Common steps in model-based methods for articulated human body pose estimation using multiview input. The dashed boxes mean that some methods may or may not have all of these steps.

multiview data. In the following sections, we will discuss in more detail the factors mentioned above.

2.2.1 Using 2-D versus 3-D Features From Multiview

Among multiview approaches, some methods use 3-D features reconstructed from multiple views [11, 16, 17, 20, 19, 27, 89, 134], e.g., volumetric (voxel) data, while others still use 2-D features [59, 71, 113], e.g., color, edges, silhouette. Since the real body pose is in 3-D, using voxel data can help avoid the repeated projection of 3-D body model onto the image planes to compare against the extracted 2-D features. Furthermore, reconstructed voxel data help to avoid the image scale issue. These advantages of using voxel data allow the design of simple algorithms, and we can make use of our knowledge about shapes and sizes of body parts. For example, Mikic et al. [89] used specific information about the shape and size of the head and torso to have a hierarchical growing procedure (detecting head first, then torso, then limbs) for body model acquisition that can be used effectively even when there is a large displacement between frames. Several methods are based on voxel data, which only indicates that voxel data is a strong cure for body pose estimation. Of course, there is an additional computational cost for voxel reconstruction, but efficient techniques for this task have also been developed [11, 20, 19, 128].

2.2.2 Tracking-based versus Single Frame-based Approaches

The modeling and tracking steps can be considered as a mapping from input space of voxel data Y and information in the predefined model (e.g., kinematic constraints) C to the body model configuration space Θ:

$$M : (Y, C) \mapsto \Theta \tag{2.1}$$

The body model configuration contains both static parameters (i.e., shape and size of each body component) and dynamic parameters (i.e., mean and orientation of each body component), in which the static parameters are estimated in the initialization step. Methods are different in the way they use and implement the mapping procedure M. Methods that have a modeling step but no tracking step are also called single frame-based methods, e.g., [134], while methods with a tracking step are called tracking-based methods, e.g., [11, 17, 24, 39, 77, 89, 140]. Because the tracker in tracking-based methods would be lost over long sequences, multiple hypotheses at each frame can be used to improve the robustness of tracking. Single frame-based approach is a more difficult issue because it does not make any assumptions on time coherence. However, we see that tracking-based methods encounter the issue of initialization or reinitialization of the tracked model.

Table 2.2: Summary of selected model-based methods for multiview body pose estimation and tracking.

Year	First Author	Data acquisition	Initialization	Body model	Method highlights	Evaluation and Precision	Real time
2001	Delamarre [27]	3 cameras	Manual	Truncated cones, spheres, parallelepipeds	Uses physical forces, a simpler form of Iterative Closest Point (ICP) to track 3-D body model from voxel data. Kalman Filter tracking.	Visual only (qualitative)	N/A
2003	Mikic [89]	6 cameras	Automatic	Ellipsoidal, cylindrical 3-D body model	Hierarchically growing procedure for initialization from head to torso, to limbs. Uses Extended Kalman Filter to predict next pose then update using growing procedure and Bayesian networks.	Visual only (qualitative)	N/A
2003	Cheung [19]	8 cameras	Automatic	Skeletal body model	Uses Colored Surface Point (CSP). Hierarchical segmentation and SFS alignment to recover motion, shape, and joint.	Synthesized ground truth. Joint position error $\sim$ 2cm	N/A
2006	Ziegler [169]	4 cameras	Manual	3-D skeletal upper body model	Reconstruct 3-D voxel data.Using ICP algorithm integrated with an Unscented Kalman Filter (UKF) to track 3-D upper body motion.	Manual label ground truth. Joint angle error $\sim 20^0$	1 fps (frame per sec)

Continued on next page

Table 2.2 – *Continued from previous page*

2007	Cheng [17]	4 cameras	Manual	Ellipsoidal 3-D body model and Gaussian representation	Integrate kinematic constraints in Kinematically Constrained Gaussian Mixture Model (KC-GMM). Derive EM algorithm with KC-GMM for pose estimation (no additional projection step).	Joint position error ∼0.5cm for synthesized hand data, ∼17cm for HumanEva-II body data	N/A
2008	Caillette [11]	3 - 5 cameras	Manual	Skeletal body model and Gaussian blobs	Break complex movement into basic motions Use Variable Length Markov Model (VLMM) to predict candidate pose. Use colored voxel for more robust tracking. Limited to a prior training motion model.	Joint position error ∼2cm for a reported sequence	5 - 28 fps
2008	Sundaresan [134]	4 - 12 cameras	Automatic	6-chain representation and super quadric 3-D body model	Segment voxel data in Laplacian Eigenspace (LE). Probabilistic register segmented voxel to body parts then estimate skeletal and superquadric parameters. No tracking.	Synthesized data (joint angle error $\sim 5^0$, nonpublic dataset, and HumanEva-II (loss of track)	N/A
2009	Tran [140]	2 cameras (wide baseline)	Automatic	3-D skeletal upper body model	Track head and head blobs. Only use 3-D head and hands movements to infer corresponding the whole upper body movement as an inverse kinematics problem.	Nonpublic dataset and HumanEva-I. Joint position error ∼10cm	15 fps

Continued on next page

Table 2.2 – *Continued from previous page*

2009	Bernier [10]	Stereo	Automatic	3-D skeletal upper body model	Use a graphical model to decompose the full 3-D pose state space into individual limb state space, coupled with a fast Nonparametric Belief Propagation for articulated pose tracking.	Synthesized data. Joint position error $\sim$7cm	10 fps
2010	Gall [39]	4 cameras	Manual	3-D surface mesh model	Multilayer framework combining global optimization, smoothing, and local optimization to improve silhouette segmentations. Track 3-D pose with ICP.	HumanEva-II dataset. Joint position error $\sim$5cm	N/A
2010	Corazza [24]	4 - 12 cameras	Automatic	3-D surface mesh model	Automatic pose-shape registration based on a database of human body shapes. Use ICP for 3-D pose tracking.	HumanEva-II and nonpublic dataset. Joint position error 1.5 - 8cm	N/A
2010	Li [77]	3 cameras	Manual	Truncated cone 3-D body model	Learn a low-dimensional manifold of human body pose from motion capture data using coordinated mixture of factor analyzer. Use Bayesian framework to track 3-D human body.	HumanEva-I dataset. Joint position error $\sim$7cm	N/A
2011	Hofmann [49]	3 cameras	Automatic	Superquadrics 3-D body model	Single-frame pose recovery with 2-D pose exemplar generation. Select 3-D pose candidates with Bayesian framework. Refine with temporal integration and model texture adaptation.	HumanEva-I and nonpublic dataset in complex environment. Joint position error $\sim$10cm	N/A

2.2.3 Manual versus Automatic Initialization

Some methods have automatic initialization step like [19, 49, 89, 134, 140] while others require a priori known or manually initialized static parameters, e.g., [17, 27, 39, 77, 169]. In [89], the specific shape and size of the head was used to design a hierarchical growing procedure for initialization. In [24], a database of human body shapes was used for initial pose-shape registration. In [10, 140], the user was asked to start at a specific pose (e.g., stretch pose) to aid the automatic initialization. In [134], Sundaresan et al. discovered an interesting property of Laplacian Eigenspace (LE) transformation: By mapping into high-dimensional (e.g., 6D) LE, voxel data of body chains like limbs, which have their length greater than their thickness, will form a 1-D smooth curve which can then be used to segment voxel data into different body chains. They then use a spline-fitting process to segment the curves which results in the segmentation of their respective body chains. This is however a single frame-based approach. The segmented voxel clusters are then registered to their actual body chain using a probabilistic registration procedure at each frame. Their results seem to be sensitive to noise in the voxel data (loss of track in the test with the public HumanEva-II dataset).

On the other hand, the Kinematically Constrained Gaussian Mixture Model (KC-GMM) method proposed by Cheng and Trivedi [17] is a tracking-based method and showed good results on the HumanEva-II dataset (won the first prize in the Workshop on Evaluation of Articulated Human Motion and Pose Estimation - CVPR EHuM2 2007 competition). However, it requires a careful manual initialization. An framework combining KC-GMM method and LE-based voxel segmentation was proposed in [141] for a more powerful human body modeling and tracking system. The LE-based voxel segmentation was used to fill in the gap of an automatic initialization of KC-GMM method. With regard to the LE-based method, combining with a tracking-based method like KC-GMM instead of doing voxel segmentation at every frame helps to overcome the sensitization to voxel noise to some extent.

2.2.4 Generic versus Application-specific Approaches for Efficiency

Depending on applications, human pose tracking may focus on different body parts including full body pose, upper body pose [92, 111], hand pose [32], and head pose [96]. Due to the complexity of human body pose estimation task, there are trade-offs in developing a generic approach versus an approach integrated to some specific cases for efficiency. For example, the KC-GMM method [17] is for generic purpose and was applied successfully for both HumanEva-II body data and synthesized hand data. However, this method is not real time because of a required manual initialization step and related computational cost. For efficiency, some methods are designed for application specific. For example, [140, 10] focus on situations in which most of the influential information of body motion is carried by the upper body and arms while the user typically sits in a fixed position. These situations arise in several realistic applications such as driver activity analysis and user activity analysis in a smart teleconference or meeting room. In [140], the problem of upper body pose tracking is broken into two subproblems: First, track the extremities including head and hand blobs. Then the 3-D movements of head and hands are used to infer the corresponding upper body movements as an inverse kinematics problem. Since the head and hand regions are typically well defined and undergo less occlusion, tracking is more reliable. Moreover by breaking the high-dimensional search problem of upper body pose tracking into two subproblems, the complexity is reduced considerably to achieve real-time performance. However, they need to deal with possible ambiguity due to the kinematic redundancy of the body model.

Another type of approaches for efficiency is to use a prior motion model from training sequences. Some representative approaches using prior motion models are [11] learning prior motion model with Variable Length Markov Model (VLMM), which can explain high-level behaviors over a long history, or [77] using a coordinated mixture of factor analyzer to learn the prior model. Compared to approaches for generic body motions [17, 24, 39, 49, 89], these approaches use the prior motion models to reduce the search space for a more efficient and robust pose tracking. However, the downside is that these methods are limited to the type of motions in training data (i.e., have difficulties if there are “unseen” movements).

2.3 Multiview Human Action Recognition

In this section, we review and compare multiview approaches for human action recognition. First, we will give an outline of approaches which solely apply 2-D multiview image data, followed up by full 3-D-based approaches, and then a description of publicly available multiview datasets as well as a comparison of several promising methods based on evaluations on the IXMAS dataset [161] and i3DPost dataset [42].

Table 2.3: Publications on multiview human action recognition.

Year	First author	Dim	Feature/Representation	Classifier/Matching	Other techniques
2005	K. Huang [54]	3-D	3-D shape context	Hidden Markov model	Tracking
2006	Canton-Ferrer [12]	3-D	3-D motion descriptors	Bayesian classifier	Ellipsoid body model
2006	Pierobon [109]	3-D	Cylindrical shape descriptor	Template matching	Dynamic time warping
2007	Lv [83]	2-D	Shape context of 2-D poses graph model	Viterbi algorithm	Synthetic training data
2007	Weinland [162]	2-D	3-D exemplars, silhouette projections	Hidden Markov model	3-D learning, 2-D classification
2008	Farhadi [34]	2-D	Hist. of silhouette and optic. flow	A transferable activity model	Cross-view recognition
2008	Junejo [66]	2-D	Self-similarity matrix descriptors	Support vector machines	Cross-view recognition
2008	Liu [78]	2-D	Spin images	Fiedler embedding	Graph-based
2008	Liu [79]	2-D	Spatiotemporal interest points	Support vector machines	Max. of mutual info.
2008	Souvenir [132]	2-D	$\mathcal{R}$ transform surfaces, manifold learning	2-D diffusion distance metric	64 virtual camera views
2008	D. Tran [144]	2-D	Motion context	Nearest neighbor	Reject unfamiliar samples
2008	Turaga [151]	3-D	Motion history volumes, Stiefel and Grassmann manifolds	Procrustes distance metric	Statistical modeling
2008	Vitaladevuni [156]	2-D	Ballistic dynamics	Bayesian model	Motion history image
2008	Yan [166]	3-D	Spatiotemporal Volumes (STV)	Maximum likelihood	Local STV features
2009	Gkalelis [43]	2-D	Multiview posture masks	Mahalanobis distance	Fuzzy vector quantization, LDA

Continued on next page

Table 2.3 – *Continued from previous page*

2009	Kilner [70]	3-D	Shape similarity	Markov model	Sports broadcast
2009	Reddy [120]	2-D	Feature-tree of Cuboids	Local voting strategy	
2010	P. Huang [57]	3-D	Shape-flow descriptor	Similarity matrix	Shape histogram
2010	Iosifidis [61]	2-D	Multiview binary masks	Mahalanobis distance	Fuzzy vector quantization, LDA
2010	Weinland [160]	2-D	3-D Hist. of Oriented Gradients	Hierarchical classification	Local occlusion handling
2011	Haq [47]	2-D	Dynamic scene geometry	Multibody fundamental matrix	Epipolar geometry
2011	Holte [51]	3-D	3-D optic. flow, harmonic motion context	Normalized correlation	Spherical harmonics
2011	Junejo [67]	2-D	Temporal self-similarities descriptors	Support vector machines	dynamic time warping (cross-view)
2011	Liu [80]	2-D	Bog of Cuboids features	Graph matching	View knowledge transfer
2011	Pehlivan [107]	3-D	Circular body layer features	Nearest neighbor	Combining pose descriptors
2011	Song [131]	3-D	HOG features	Hidden conditional random fields	Body and hand movements

2.3.1 2-D Approaches

One line of work concentrates solely on the 2-D image data acquired by multiple cameras. Action recognition can range from pointing gesture to complex multisignal actions, e.g., including both coarse level of body movement and fine level of hand gesture.

1. Shape and Silhouette Features: In the work of Souvenir et al. [132], the acquired data from 5 calibrated and synchronized cameras, is further projected to 64 evenly spaced virtual cameras used for training. Actions are described in a view-invariant manner by computing $\mathcal{R}$ transform surfaces of silhouettes and manifold learning. Gkalelis et al. [43] exploit the circular shift invariance property of the Discrete Fourier Transform (DFT) magnitudes and use Fuzzy Vector Quantization (FVQ) and Linear Discriminant Analysis (LDA) to represent and classify actions from multiview silhouettes. Another approach is proposed by Iosifidis et al. [61], where binary body masks from frames of a multicamera setup used to produce the i3DPost Multiview Human Action Dataset [42], are concatenated to multiview binary masks. These masks are rescaled and vectorized to create feature vectors in the input space. FVQ is performed to associate input feature vectors with movement representations, and LDA is used to map movements in a low-dimensionality discriminant feature space.

2. Motion Features: Some authors perform action recognition using motion features or a combination of static shape and motion features from image sequences in different viewing angles. Ahmad et al. [3] apply Principal Component Analysis (PCA) of optical flow velocity and human body shape information and then represent each action and viewpoint using a set of multidimensional discrete Hidden Markov Models (HMM). Matikainen et al. [84] propose a method for multiuser, prop-free pointing detection using two camera views. The observed motion are analyzed and used to refer the candidates of pointing rotation centers and then estimate the 2-D pointer configurations in each image. Based on the extrinsic camera parameters, these 2-D pointer configurations are merged across views to obtain 3-D pointing vectors. Cherla et al. [18] show how view-invariant recognition can be performed by using data fusion of two orthogonal views. An action basis is built using eigenanalysis of walking sequences of different people, and projections of the width profile of the actor and spatiotemporal features are applied. Finally, Dynamic Time Warping (DTW) is used for recognition.

3. Synthetic Training Data: Others use synthetic data rendered from a wide range of viewpoints to train their model and then classify actions in a single view, e.g., Lv et al. [83], where shape context is applied to represent key poses from silhouettes and Viterbi Path Searching for classification. A similar approach was proposed by Fihl et al. [36] for gait analysis.

4. Cross-view Recognition: Another topic which has been explored by several authors the last couple of years is cross-view action recognition. This is a difficult task of recognizing actions by training on one view and testing on another completely different view (e.g., the side view versus the top view of a person in IXMAS). A number of techniques have been proposed, stretching from applying multiple features [78], information maximization [79], dynamic scene geometry [47], self-similarities [66, 67] and transfer learning [34, 80].

5. Other Techniques: A number of other techniques have been employed, like metric learning [144] or representing action by feature-trees [120] or ballistic dynamics [156]. In [160], Weinland et al. propose an approach which is robust to occlusions and viewpoint changes using local partitioning and hierarchical classification of 3-D Histogram of Oriented Gradients (3-DHOG) volumes. For additional related work on view-invariant approaches, please refer to the recent survey by Ji et al. [63].

2.3.2 3-D Approaches

Another line of work utilizes the full reconstructed 3-D data for feature extraction and description. Figure 2.2 shows some examples of the more prominent model and non-model-based representations of the human body and its motion. These will be reviewed below along with a number of other recent 3-D approaches.

1. 3-D Shape and Pose Features: Johnson and Hebert proposed the spin image [65], and Osada et al. the shape distribution [103]. Ankerst et al. introduced the shape histogram [5], which is a similar to the 3-D extended shape context [8] presented by Körtgen et al. [72], and Kazhdan et al. applied spherical harmonics to represent the shape histogram in a view-invariant manner [68]. Later Huang et al. extended the shape histogram with color information [56]. Recently, Huang et al. made a comparison of these shape descriptors combined with self-similarities, with the shape histogram (3-D shape context) as the top-performing descriptor [57].

2. Temporal Information and Alignment: A common characteristic of all these approaches is that they are solely based on static features, like shape and pose description, while the most popular and best-performing 2-D image descriptors apply motion information or a combination of the two [92, 163]. Some authors add temporal information by capturing the evolvement of static descriptors over time, i.e., shape and pose changes [12, 23, 54, 70, 109, 161, 162, 166].

The common trends are to accumulate static descriptors over time, track human shape or pose information, or apply sliding windows to capture the temporal contents [92, 109, 161, 163]. Recently, Huang et al. proposed 3-D shape matching in temporal sequences by time filtering and shape flows [57]. Kilner et al. [70] applied the shape histogram and evaluated similarity measures for action matching and key-pose detection in sports events, using 3-D data available in the multicamera broadcast environment. Cohen et al. [23] used 3-D human body shapes and Support Vector Machines (SVM) for view-invariant identification of human body postures. They apply a cylindrical histogram and compute an invariant measure of the distribution of reconstructed voxels, which later was used by Pierobon et al. [109] for human action recognition. Another example is seen in the work of Huang and Trivedi [54], where a 3-D cylindrical shape context is presented to capture the human body configuration for gesture analysis of volumetric data. The temporal information of an action is modeled using HMM. However, this study does not address the view-independence aspect. Instead, the subjects are asked to rotate while training the system.

Pehlivan et al. [107] presented a view-independent representation based on human poses. The volume of the human body is first divided into a sequence of horizontal layers, and then the intersections of the body segments with each layer are coded with enclosing circles. The circular features in all layers - (i) the number of circles, (ii) the area of the outer circle, and (iii) the area of the inner circle - are then used to generate a pose descriptor. The pose descriptors of all frames in an action sequence are further combined to generate corresponding motion descriptors. Action recognition is then performed with a simple nearest neighbor classifier.

3. Model-based Human Pose Tracking: More detailed 3-D pose information (i.e., from tracking the kinematics model of the human body) is a rich and view-invariant representation for action recognition but challenging to derive [112]. Human body pose tracking is itself an important area with many related research studies. Among these, research started with monocular view and 2-D features, and more recently (about 10 years ago) multiview and 3-D features like volumetric data have been applied for body pose estimation and tracking [142]. One of the earliest methods for multiview 3-D human pose tracking using volume data was proposed by Mikic et al. [89], in which they use a hierarchical procedure starting by locating the head using its specific shape and size, and then growing to other body parts. Though this method showed good visual results for several complex motion sequences, it is also quite computationally expensive. Cheng and Trivedi [17] proposed a method that incorporates the kinematics constraints of a human body model into a Gaussian Mixture Model framework, which was applied to track both body and hand models from volume data. Although this method was highly rated with good body tracking accuracy on HumanEva dataset [127], it requires a manual initialization and could not run in real time. We see that there are always trade-offs between achieving detailed information of human body pose and the computational cost as well as the robustness. In [131], Song et al. focus on gestures with more limited body movements. Therefore, they only use the depth information from two camera views to track 3-D upper body poses using a Bayesian inference framework with a particle filter, as well as classifying several hand poses based on their appearance. The temporal information of both upper body and hand pose are then inputted into a Hidden Conditional Random Field (HCRF) framework for aircraft handling gesture recognition. To deal with the long range temporal dependencies in some gestures, they also incorporate a Gaussian temporal smoothing kernel into the HCRF inference framework.

4. 3-D Motion Features: The Motion History Volume (MHV) was proposed by Weinland et al. [161], as a 3-D extension of Motion History Images (MHIs) (see Figure 2.2). MHVs are created by accumulating static human postures over time in a cylindrical representation, which is made view-invariant with respect to the vertical axis by applying the Fourier transform in cylindrical coordinates. The same representation was used by Turaga et al. [151] in combination with a more sophisticated action learning and classification based on Stiefel and Grassmann manifolds. Later, Weinland et al. [162] proposed a framework, where actions are modeled using 3-D occupancy grids, built from multiple viewpoints, in an exemplar-based Hidden Markov Models (HMM). Learned 3-D exemplars are used to produce 2-D image information which is compared to the observations; hence, 3-D reconstruction is not required during the recognition phase. Canton-Ferrer et al. [12] propose another view-invariant representation based on 3-D MHIs and 3-D invariant statistical moments. A different strategy is presented by Yan et al. [166]. They propose a 4D action feature model (4D-AFM) for recognizing actions from arbitrary views based on spatiotemporal features of spatiotemporal volumes (STVs). The extracted features are mapped from the STVs to a sequence of reconstructed 3-D visual hulls over time, resulting in the 4D-AFM model, which is used for matching actions. A 3-D descriptor which is directly based on rich detailed motion information is the 3-D Motion Context (3-D-MC) [51] and the Harmonic Motion Context (HMC) [51] proposed by Holte et al. The 3-D-MC descriptor is a motion-oriented 3-D version of the shape context [8, 72], which incorporates motion information implicitly by representing estimated 3-D optical flow by embedded Histograms of 3-D Optical Flow (3-D-HOF) in a spherical histogram. The HMC descriptor is an extended version of the 3-D-MC descriptor that makes it view-invariant by decomposing the representation into a set of spherical harmonic basis functions.

2.3.3 Multiview Datasets

A number of multiview human action datasets are publicly available. A frequently used dataset is the INRIA Xmas Motion Acquisition Sequences (IXMAS) Multiview Human Action Dataset [161].[1] It consists of 12 nonprofessional actors performing 13 daily-life actions 3 times: *check watch*, *cross arms*, *scratch head*, *sit down*, *get up*, *turn around*, *walk*, *wave*, *punch*, *kick*, *point*, *pick up*, and *throw*. The dataset has been recorded by 5 calibrated and synchronized cameras, where the actors chose freely position and orientation, and consists of image sequences (390×291) and reconstructed 3-D volumes ($64 \times 64 \times 64$ voxels), resulting in a total of $2,340$ action instances for all 5 cameras. Figure 2.5 shows multiview actor/action images and voxel-based volume examples from the IXMAS datasets.

Recently, a new high-quality dataset has been produced, the i3DPost Multiview Human Action and Interaction Dataset [42].[2] This dataset, which has been generated within the Intelligent 3-D Content Extraction and Manipulation for Film and Games EU funded research project, consists of 8 actors performing 10 different actions, where 6 are single actions: *walk*, *run*, *jump*, *bend*, *hand-wave* and *jump-in-place*, and 4 are combined actions: *sit-stand-up*, *run-fall*, *walk-sit*, and *run-jump-walk*. Additionally, the dataset also contains 2 interactions: *handshake* amd *pull*, and 6 basic facial expressions. The subjects have different body sizes, clothing and

[1] The IXMAS dataset is available at http://4drepository.inrialpes.fr/public/viewgroup/6

[2] The i3DPost dataset is available at http://kahlan.eps.surrey.ac.uk/i3DPost_action/data

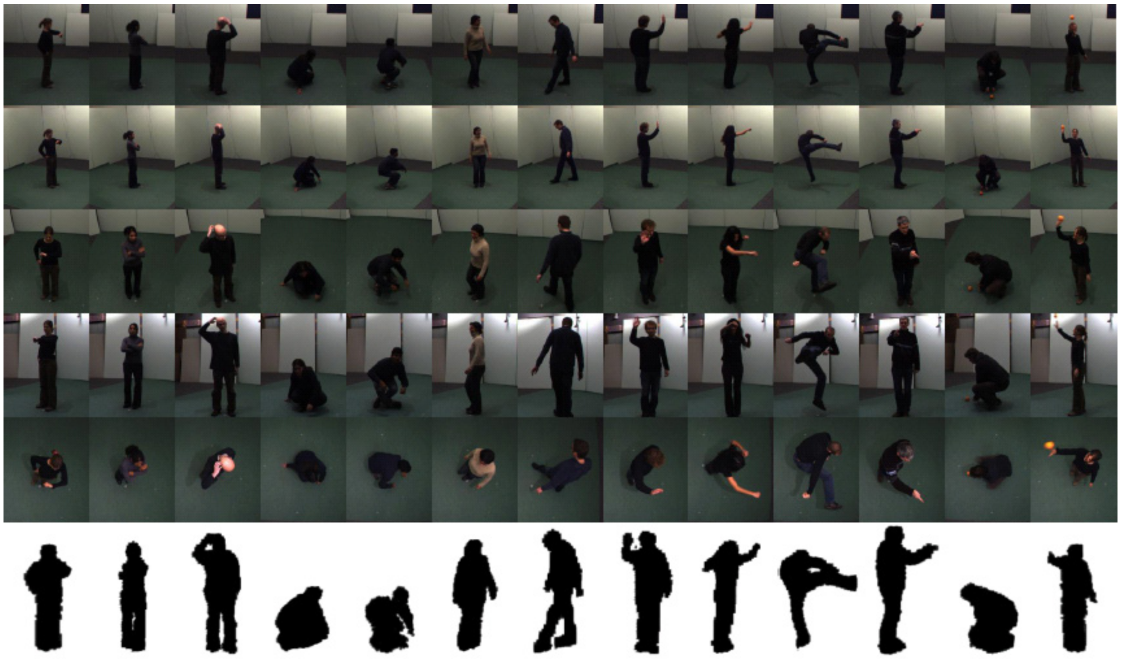


Figure 2.5: Image and 3-D voxel-based volume examples for the 13 actions from the IXMAS Multiview Human Action Dataset. The figure is organized such that the columns correspond to the 13 different actions performed by the 12 actors. The first 5 rows depict images captured from the 5 camera views, while the 6$^{\text{th}}$ row shows the corresponding 3-D volumes.

are of different sex and nationalities. The multiview videos have been recorded by a 8 calibrated and synchronized camera setup in high-definition resolution (1920 × 1080), resulting in a total of 640 videos (excluding videos of interactions and facial expressions). For each video frame, a 3-D mesh model of relatively high detail level ($20,000$ - $40,000$ vertices and $40,000$ - $80,000$ triangles) of the actor and the associated camera calibration parameters are available. The mesh models were reconstructed using a global optimization method proposed by Starck and Hilton [133]. Figure 2.6 shows multiview actor/action images and 3-D mesh model examples from the i3DPost dataset.

Another interesting multiview dataset is the Synchronized Video and Motion Capture Dataset for Evaluation of Articulated Human Motion (HumanEva) [127], containing 6 simple actions performed by 4 actors, captured by 7 calibrated video cameras (4 grayscale and 3 color), which have been synchronized with 3-D body poses obtained from a motion capture system. Among other less frequently used multiview datasets are the CMU Motion of Body (MoBo) Database [45], the Multicamera Human Action Video Dataset (MuHAVi) [1], and the KU Gesture Dataset [60].

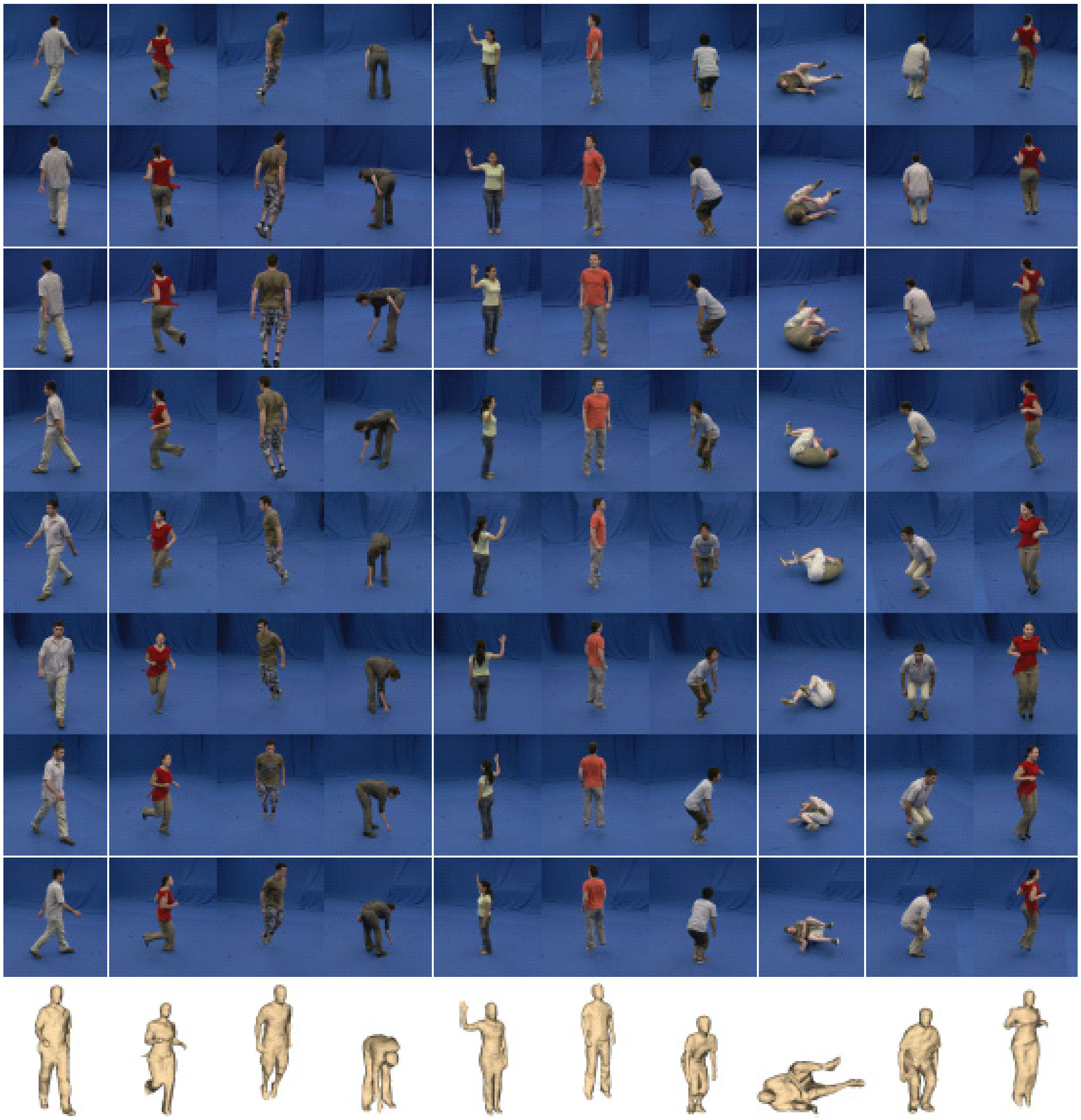


Figure 2.6: Image and 3-D mesh model examples for the 10 actions from the i3DPost Multiview Human Action Dataset. The figure is organized such that the columns correspond to the 10 different actions performed by the 8 actors, where the first 6 columns show the single actions and the last 4 columns show the combined actions. The first 8 rows depict images captured from the 8 camera views, while the 9th row shows the corresponding 3-D mesh models.

2.3.4 Comparison

In Table 2.4, the recognition accuracies of several 2-D and 3-D approaches evaluated on IXMAS are listed. It is interesting to note that all the 3-D approaches except one are the top-performing methods. Especially, the methods proposed by Turaga et al. [151] and Weinland et al. [161], which both are based on Motion History Volumes (MHVs), produce superior recognition accuracies. The approach in [151] is based on the prior work of Weinland et al., but applies a more sophisticated action learning and classification based on Stiefel and Grassmann manifolds, which leads to a significant improvement.

Another interesting approach with high performance is the work by Pehlivan et al. [107], which uses 3-D pose features represented by horizontal circular pose features over time. This shows that methods based on full 3-D shape and pose information are also promising 3-D action recognition strategies. In contrast, the work on 3-D action recognition of Yan et al. [166], where a 4D action feature model (4D-AFM) based on spatiotemporal features of spatiotemporal volumes (STVs) is developed, results in a lower recognition rate than some 2-D methods. The authors develop a 4D action feature model (4D-AFM) based on spatiotemporal features extracted from a sequence of spatiotemporal volumes (STVs). A reason might be that the low-quality multiview video data of IXMAS produces noisy sequences of reconstructed 3-D visual hulls over time, and therefore distorts the extraction of more fine-detailed features from the sequence of STVs.

The best-performing 2-D methods are the work proposed by Vitaladevuni et al. [156], Weinland et al. [160], and Liu et al. [79]. In [156], Vitaladevuni et al. use motion history images features and ballistic dynamics, where actions are represented and classified by a Bayesian model, while both [160] and [79] use local features in the form of spatiotemporal interest points to extract a bag of visual words of Cuboid features and local partitioning and hierarchical classification of 3-D Histogram of Oriented Gradients volumes, respectively. In both cases, support vector machines are applied for classification. However, Weinland et al. directly show the method's robustness to occlusions and viewpoint changes and report near real-time performance.

This evaluation indicates that the use of the full reconstructed 3-D information is superior to applying 2-D image data from multiple views, when it comes to recognition accuracy. However, the computational cost of working in 3-D is usually also more expensive. Hence, with respect to the application and demand for real-time performance, 2-D approaches might still be the best choice. It should be noted that some results are reported using cross-view evaluation, which is more challenging than applying data from multiple and identical viewpoints; however, still some of these methods perform very well. Especially, the approaches proposed by Haq et al. [47] using dynamic scene geometry and Liu et al. [80] adopting a transfer learning model give superior cross-view recognition. When both types of results are available in the original work, we have reported the results for all views, since these are more comparable to the 3-D results, where all views are used to reconstruct 3-D data.

Table 2.4: Recognition accuracies (%) for the IXMAS dataset. The column named "Dim" states if the methods apply 2-D image data or 3-D data; the other columns state how many actions are used for evaluatiom, and if the results are based on all views or cross-view recognition.

Year	Method	Dim	11 actions	13 actions	All views	Cross-view
2008	Turaga et al. [151]	3-D	98.78	-	x	
2006	Weinland et al. [161]	3-D	93.33	-	x	
2011	Pehlivan et al. [107]	3-D	90.91	88.63	x	
2008	Vitaladevuni et al. [156]	2-D	87.00	-	x	
2011	Haq et al. [47]	2-D	83.69	-		x
2010	Weinland et al. [160]	2-D	83.50	-	x	
2008	Liu et al. [79]	2-D	-	82.80	x	
2011	Liu et al. [80]	2-D	82.80	-		x
2007	Weinland et al. [162]	2-D	81.27	-	x	
2007	Lv et al. [83]	2-D	-	80.60	x	
2008	Tran et al. [144]	2-D	-	80.22	x	
2008	Cherla et al. [18]	2-D	-	80.05	x	
2008	Liu et al. [78]	2-D	-	78.50	x	
2008	Yan et al. [166]	3-D	78.00	-	x	
2011	Junejo et al. [67]	2-D	74.60	-	x	
2008	Junejo et al. [66]	2-D	72.70	-	x	
2009	Reddy et al. [120]	2-D	-	72.60	x	
2008	Farhadi et al. [34]	2-D	58.10	-		x

Table 2.5: Recognition accuracies (%) for the i3DPost dataset. *Gkalelis et al. [43] test on 5 single actions.

Year	Method	Dim	8 actions
2011	Holte et al. [51]	3-D	92.19
2010	Iosifidis et al. [61]	2-D	90.88
2009	Gkalelis et al. [43]	2-D	90.00*

Table 2.5 shows the recognition accuracies of a few other approaches evaluated on the i3DPost dataset. The evaluation has been carried out for 8 actions by combining the 6 single actions in the dataset with two additional single actions: sit down and fall by splitting 2 of the 4 combined actions. Again the approach based on full 3-D motion information in the form of 3-D optical flow and Harmonic Motion Context (HMC) by Holte et al. [51] outperforms the 2-D methods by Gkalelis et al. [43] and Iosifidis et al. [61], which both use shape features from multiview body masks of 2-D human silhouettes, Fuzzy Vector Quantization, and Linear Discriminant Analysis. This strengthens the similar outcome of the more extensive comparison using IXMAS. Generally, the top-performing approaches for the two datasets are the 3-D methods based on 3-D motion features by Turaga et al. [151], Weinland et al. [161], and Holte et al. [51]. However, it should be noted that all these methods for human action recognition are basically model-free, which means that they do not apply a specific human body model to model and estimate the exact position and configuration of the body parts and joints. Hence, these methods are only applicable for a set of the applications in Table 2.1. This results in a need for model-based approaches for 3-D pose estimation and exact modeling of the human body.

2.4 Discussion

In this section, we provide a review and comparative study of recent developments for human pose estimation and activity recognition using multiview data. We give an overview of the different application areas and their associated requirements for a successful operation.

First, we review the subarea of model-based methods for real human body pose estimation using volumetric data. After a brief overview to put the topic into context, we focus on analyzing and comparing several selected methods, especially some recent methods proposed in the past two years, to highlight their important results. This includes: increasing generality, real-time performance, and a new general LE-based method for voxel segmentation. There are some related open-ended research areas that should be mentioned. First is the issue of human body pose estimation at multilevel (e.g., body level, head level, and hand level) which was mentioned in [145]. We can see the benefits of having such a multilevel human body pose estimation system, such as: combined information from different level of details is more useful (e.g., in intelligent environment, the combination of body pose, hand pose, and head pose would give better interpretation of human status/intention); information from different levels can complement each other and help to improve the estimation performance. However, typical approaches in this area deal with each of these tasks (body pose estimation, hand pose estimation, head pose estimation, etc.), separately. Therefore, it is useful to conduct further studies to analyze the reasons why typical approaches only deal with one task at a time, and find a way to achieve the goal of a full body model (e.g., including body, head, and hand). Another related open-ended research area that is important, is the issue of pose estimation and tracking of multiple objects simultaneously.

Next, the subarea of multiview action recognition is reviewed, covering both 2-D and 3-D multiview approaches, and publicly available multiview datasets. A qualitative comparison of several promising approaches based on the IXMAS and i3DPost datasets, reveals that methods using 3-D representations of the data turn out to outperform the 2-D methods. The main strength of multiview setups is the high-quality full-volume 3-D data, which can be provided from 3-D reconstruction by shape-from-silhouettes and refinements techniques. It also helps to uncover occluded action regions from different views in the global 3-D data and allows for extraction of informative features in a more rich 3-D space, than the one captured from a single view. However, although the reviewed approaches show promising results for multiview human pose estimation and action recognition, 3-D reconstructed data from multiview camera systems has some shortcomings. First of all, the quality of the silhouettes is crucial for the outcome of applying shape-from-silhouettes. Hence, shadows, holes and other errors due to inaccurate foreground segmentation will affect the final quality of the reconstructed 3-D data. Secondly, the number of views and the image resolution will influence the level of details which can be achieved, and self-occlusion is a known problem when reconstructing 3-D data from multiview image data, resulting in merging body parts. Finally, 3-D data can only be reconstructed in a limited space where multiple camera views overlap.

In recent years, other prominent vision-based sensors for acquiring 3-D data have been developed. Time-of-Flight (ToF) range cameras, which are single sensors capable of measuring depth information, have become popular in the computer vision community. Especially, with the introduction of the Microsoft Kinect sensor [125], these single and direct 3-D imaging devices have become widespread and commercial available at a low cost. Their applicability are broader due to the convenience of using a single sensor, avoiding the difficulties inherent to classical stereo and multiview approaches (the correspondence problem, careful camera placement, and calibration). However, in contrast to the rich full-volume 3-D data which can be provided by 3-D reconstruction from multiview data, these sensors only capture 3-D data of the frontal surfaces of humans and other objects. Additionally, these sensors are usually limited to a range up to about 6 - 7 meters, and the estimated data can become distorted by scattered light from reflective surfaces.

2.5 Acknowledgments

Chapter 2 is based on material that is published in the IEEE Journal of Selected Topics in Signal Processing (2012) by Michael B. Holte, Cuong Tran, Mohan M. Trivedi, and Thomas B. Moeslund. The dissertation author was a major investigator and author of this paper ($\sim 50\%$ text contribution).

Chapter 3

Human Body Pose Modeling and Tracking at Different Levels of Detail

In this chapter, we develop some approaches for human body pose modeling and tracking at different levels of detail starting with an integrated framework with automatic initialization for body and hand pose modeling and tracking from 3-D voxel data. Considering the trade-offs in generic versus application-specific approaches to achieve the efficiency and robustness required for interactive applications, we also develop an eXtremity Movement OBservation (XMOB) framework for 3-D upper body pose tracking in real time, which works well for situations where the arms carry the most influential information of body motion (e.g., in meeting room, teleconference, and driver assistance situations). In the last section of this chapter, we will discuss our initial work in multilevel human pose modeling and tracking by combining the outputs from different body levels.

3.1 Human Pose Modeling and Tracking from Voxel Data: An Integrated Framework with Automatic Initialization

Human pose modeling and tracking is an important research area with many potential applications including advance Human Computer Interaction (HCI), surveillance, 3-D animation, intelligent environment, robot control, etc. Compared to previous pose estimation technologies using markers or some specific devices, markerless vision-based approaches provide more natural, noncontact solutions. This is, however, a very challenging task. One major reason is the very high dimensionality of the pose configuration space, which grows exponentially with the number of DOF (Degree Of Freedom) of the articulated model. For example, in [17], they used 19 DOF body model and 27 DOF hand model. Moreover, we also have to deal with other common issues in computer vision including self-occlusion, variation in lighting conditions, shadows, and object appearance (e.g., different clothes, and hair).

There have been numerous research studies in this area with over 350 publications during the recent period from 2000 to 2006 [90]. Some surveys of several techniques for articulated body pose estimation can be found in [90, 111, 164] for body, [32] for hand, and [96] for head. These research studies however only deal with tasks at each level (body pose, hand pose, head pose) one at a time, for example body pose estimation refers to the body with torso, head, and 4 limbs but without the detailed hand or head/facial model. Nevertheless, we see that the ability to estimate a multilevel full body model (e.g., the model consisting of body, hands, and head) is desirable: First, the combined information from multiple levels (body, hand, head) is useful. For example, some recent researches in intelligent environment proposed using gait in addition to face only for better people recognition and/or human affect recognition (e.g., [46]). Second, information from different levels can support each other and help to improve the estimation performance. For this reason, we propose a framework for estimating human pose at multiple levels in an integrated way and then combining the results from each level into a full model of body, hand, and head (in Section 3.3).

Among different approaches to the problem of human body pose estimation, we focus on model-based methods that use reconstructed voxel data and aim to estimate real 3-D human pose. Based on the results from KC-GMM method [17] and LE-based method [134], we propose a fairly general method combining automatic initialization and modeling which was applied for both body and hand pose estimation. In our experiment, this combined method was shown more powerful than using the methods in [17] and [134] solitarily.

3.1.1 Related Research Studies

Voxel data is normally a binary 3-D matrix V representing a predefined space of interest where we expect to see the human body. A voxel in V with value 1 means that that voxel belongs to the concerned subject. The output of 3-D estimated human body pose X contains information about the absolute position (μ) and orientation (rotation matrix R) of each body component $X = \{(\mu_i, R_i)\}_{i=1:n}$. A more convenient way to represent X is the twist framework [89], which uses a global position and orientation (e.g., the position and orientation of torso) and a sequence of relative angles between body parts in a hierarchical body tree (e.g., torso → upper arm → lower arm).

The advantages of using the twist framework are as follows: First, it results in a nonredundant set of model parameters. Second, the relative angles are easier for humans to intuit than the absolute position and angles. Finally, most of the constraints that ensure physically valid body configurations are now inherent to the twist model.

In a typical flowchart of common model-based pose estimation methods using voxel data (Figure 2.4), there are five main steps:

- Camera calibration/Data capture
- Voxel reconstruction
- Initialization/Segmentation (segment voxel data into different body parts)
- Modeling/Estimation (estimating pose using current frame only)
- Tracking (use temporal information from previous frames in estimating body pose in current frame)

The first two steps are common for all methods of this kind while among the last three steps, different methods may touch different combinations of these steps as discussed later.

In data capture and voxel reconstruction steps, a common vision-based approach is the shape-from-silhouette (visual hull). First, the images from multiple synchronized cameras are segmented into object silhouette using some background subtraction techniques [31, 53]. Then some efficient shape-from-silhouette techniques [19, 128] can be used to retrieve the 3-D voxel data. There is also another approach called shape-from-photo consistency (photo hull) [129] that uses other features such as color and edges (not just the silhouette) from the photos to have more accurate geometry of the reconstructed hull. In case of human body pose estimation, the more accurate geometry of voxel data is not necessary, so using visual hull is more appropriate because it should be faster and more robust to noise (i.e., noise caused by detailed features like edges).

The modeling and tracking steps can be considered as a fitting (mapping) procedure to find the optimal fit (optimal pose) between the predefined body model and the observed voxel data. Methods are different in the choice of body models and the mapping procedure (the last three steps in Figure 2.4). Figure 2.2 shows some examples of used body models including skeletal model, model with ellipsoidal components, model using superquadric representation, and model with Gaussian blobs representation. In the following paragraphs, we will discuss in more details the steps of the mapping procedure.

The body model configuration contains both static parameters (i.e., shape and size of each body component) and dynamic parameters (i.e., position and orientation of each body component), in which the static parameters are estimated in the initialization step, using information in the predefined model. Many methods require a manual initialization [11, 15, 17, 27, 152] while some methods have automatic initialization step like the hierarchical growing procedure for body model acquisition in [89]. This method is therefore fully automated; however, because of using specific information of the shape and size of body parts, it lacks generality to be applied to other articulated body model like hand. Werghi et al. [164] also proposed a topological analysis framework for segmenting 3-D human body data into functional body parts (5 parts including 4 limbs and torso+head). Their experiment showed the ability of the proposed method in coping with various body postures and noise. This paper however stopped at this coarse segmentation result, and no finer processing and analysis were developed. A more recent method using the discovered property of Laplacian Eigenspace (LE) mapping for automatic initialization of body model was introduced by Sundaresan et al. [134]. After mapping into high-dimensional LE, voxel data of body parts with their length greater than their thickness will form a smooth 1D curve. A spline-fitting is then used to segment body voxel into different long chain parts (head, torso, and 4 limbs). This method can handle poses where there is self-contact, i.e., when one or more limbs touch other body parts and it is fairly general (although in [134] they used it for body model only, this method can be applied to other articulated models constructed of long chains). Their experiment with HumanEva-II dataset however indicates that the LE-based voxel segmentation is quite sensitive to voxel noise and fails when limbs are not well separated and noise voxel can be segmented as ghost limbs.

Some methods may have the modeling step but no tracking step. These methods are called single frame-based methods because only information in current frame is used to estimate body pose. By contrast, methods with tracking step that make use of temporal/dynamic information from previous frames to estimate current pose are called tracking-based methods. A popular technique for tracking is the Kalman Filter, e.g., [27], or Extended Kalman Filter (with nonlinear transition and measurement equations), e.g., [89], for pose prediction. However, for faster and more complex nonlinear movement prediction, a better predictive scheme is required. Caillette et al. [11] proposed a predictive scheme in which complex human activities such as dancing are broken into elementary movements by clustering a feature space constructed as follows: with each pose, the corresponding feature vector $F_t = (x_t, x'_t)$ consisting of joint angle vector x_t without global position and orientation (to avoid sensitivity to this kind of global information) and its first derivative x'_t (to help in resolving "velocity" ambiguities). After clustering the feature space, the pose prediction within a cluster (an elementary movement) can be done more efficiently and accurately. Besides the local dynamic within a cluster, they proposed using a Variable Length Markov Model (VLMM) for possible transitions between clusters. Different from fixed-order Markov Model, VLMM can fit a higher Markov in "needed" contexts, while using lower-order Markov elsewhere so VLMM can be used to explain high-level behaviors over a long history. These "needed" contexts are determined by the training data. This method is among very few methods that reported real-time performance; however, it requires a good training phase and cannot deal with behaviors unseen in the training data. To deal with the issue of very large training sets in discriminative approaches for human pose estimation, an online probabilistic regression scheme was also proposed [153].

The output of fitting procedure could be a global optimal solution or a local suboptimal solution. For instance, Cheng et al. [17] used the same paradigm of probabilistic clustering for pose estimation in which each body (or hand) component is considered as a Gaussian and the whole body is a mixture of Gaussians. They then integrated the kinematic constraints between body component into the Gaussian Mixture Model to have a Kinematically Constrained Gaussian Mixture Model (KC-GMM) and then derived the EM algorithm for the new model to find the optimal body pose. This method has generality, and they did experiment with both body model and hand model. However, due to the nature of EM algorithm, this method can easily get stuck at a suboptimal solution especially when there is a large displacement. In contrast, the method in [11] provides a global optimal solution by using a Monte Carlo Bayesian framework to estimate the posterior distribution of body pose given observed voxel data $P(X \mid Z)$. However because of the very high dimensionality of body pose configuration space, the "sufficient" number of particles to accurately represent the true posterior distribution $P(X \mid Z)$ could be very large, which leads to a computational issue. In [11], they addressed this pitfall and achieved real-time performance by using VLMM prediction scheme mentioned above and a Gaussian blob fitting procedure for a faster likelihood evaluation. It should be mentioned that both KC-GMM method [17] and Monte Carlo-based method [11] require a manual initialization.

3.1.2 A General Body Pose Estimation Method: Combining Initialization and Tracking

As discussed above, a desired improvement of KC-GMM method [17] is an automated initialization step (among several methods competing in the Workshop on Evaluation of Articulated Human Motion and Pose Estimation - CVPR EHuM2 2007, KC-GMM method won the first prize). A possible solution is to use hierarchical growing procedure for body model acquisition [89]. In doing so, however, we will lose the generality of KC-GMM method and cannot apply it to the hand model. The voxel segmentation using LE transformation in [134] has generality, so it would be a more appropriate choice for improving KC-GMM method with an automated initialization step. With regard to LE-based method for body modeling, we know that the voxel segmentation step is sensitive to noise, and failure in this initial step will affect their subsequent steps of skeletal model estimation and superquadric body model estimation. This motivates our idea of a combined method [141] in which instead of doing voxel segmentation at every frame, we only use it for initialization purpose. Then in subsequent frames, a tracking method like KC-GMM that exploits temporal history information could help in overcoming the sensitivity to noise to some extent. Concrete steps of the proposed combined method are shown in Figure 3.1. We assume that the body/hand starts at an specific initial pose, which clearly reveals the body/hand's structure (e.g., stretch pose). The LE transformation is then applied to segment body/hand voxel in this first frame into different parts (e.g., limbs in body model and fingers in hand model). This segmentation result is then used to fill the gap of an automated body/hand model initialization for the KC-GMM method.

Voxel Reconstruction

The implementation of voxel reconstruction came from the previous working result in [17]. The Horprasert background subtraction [53] was used to extract silhouettes from multiple camera views, then voxel data was reconstructed using the shape-from-silhouette (visual hull) technique.

LE-based Voxel Segmentation in The First Frame

In step (b), Figure3.1, we implement the LE-based voxel segmentation method described in [134] as we apply it to both body case and hand case (in [134] they only described and implemented the method for body case).

The main point of LE-based voxel segmentation is the discovered property of Laplacian Eigenspace (LE) transformation: By mapping into high-dimensional (e.g., 6D) LE, voxel data of body chains like limbs, which have their length greater than their thickness, will form a 1-D smooth curve in the LE which can then be used to segment voxel data into different body chains.

The procedure for LE mapping is as follows: First, we compute the adjacency matrix W of voxel data, such that $W_{ij} = 1$ only if voxel i is a neighbor of voxel j. Then, we compute D matrix, so that and $D_{ij} = 0$ for $i \neq j$. The first d eigenvectors of $L = D - W$ with minimum eigenvalues give us the d basis of the needed LE.

After mapping into LE, there are several 1-D curves corresponding to different body chains. A spline-fitting procedure, which starts from the farthest point in LE and then tries to grow a spline-fitting onto nearby points until the fitting error exceeds a threshold, is used to segment the curves in LE which results in the segmentation of their respective body chains voxel.

In applying LE-based voxel segmentation to hand case: because fingers also have their length greater than their thickness, they will form 1-D smooth curves in LE, and we can apply the spline-fitting to segment hand voxel data as we do with body voxel data. In case of hand, however, the palm would not form a good 1-D curve in LE due to its nearly square shape, so we only use spline-fitting to segment voxel of 5 fingers, and the remaining voxels are considered palm's voxels. The segmentation results of applying spline-fitting process in LE for both body voxel and hand voxel are shown in Figure 3.1.(b).

Automatic Model Initialization in The First Frame

In the automatic initialization step (step (c) in Figure 3.1), we have the segmented body/hand voxel and a template of body/hand model, which contains a predefined body/hand structure (i.e., the number of components, the number of joint, and the number of DOF at each

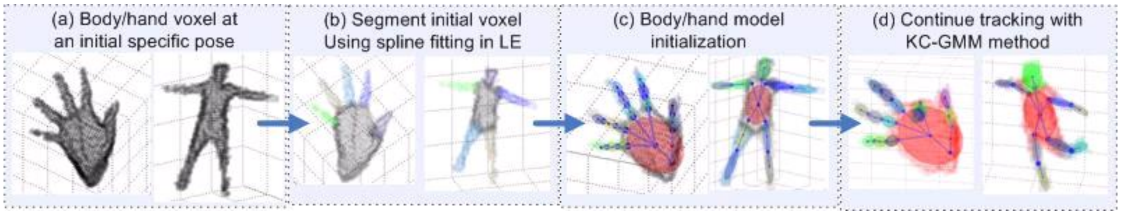

(a) Body/hand voxel at an initial specific pose
(b) Segment initial voxel Using spline fitting in LE
(c) Body/hand model initialization
(d) Continue tracking with KC-GMM method

Figure 3.1: Steps of the combined method for automatic body/hand model initialization and tracking.

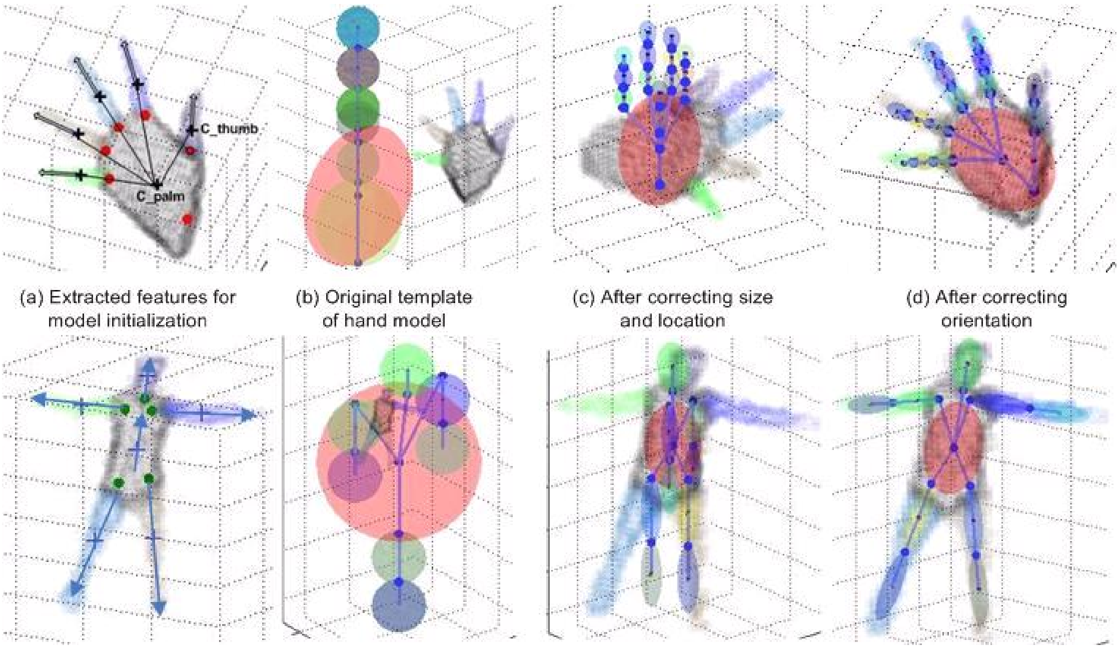

C_thumb
C_palm
(a) Extracted features for model initialization
(b) Original template of hand model
(c) After correcting size and location
(d) After correcting orientation

Figure 3.2: (a)- A simple procedure for body/hand model initialization based on segmented voxel data (b)(c)(d)-Body/hand model initialization result.

joint). Our goal is to adjust the necessary parameters including component dimensions, joint positions, and angles to fit the segmented voxel well. Because we require the body/hand to start at a specific pose (i.e., stretch pose), this initialization step can be done with the following simple and fast procedures (Figure 3.2.(a)).

Body model initialization:

- For body part registration, the center of each segmented voxel region is computed. Two lowest parts (in z-direction) are registered as the two legs. The remaining biggest part is the torso. The remaining part that is closest to the torso center in xy-plane is the head, and the other two parts are the two arms. The left/right assignment is done with the assumption that arms tend to point forward.

- The local z-axis of the torso and each limb is computed as the largest PCA component of corresponding voxel region (marked with arrows). The local z-axis of the head is set to be the same as the local z-axis of the torso. With the assumption of stretch pose, we can use the found local z-axis to compute the local x-axis and y-axis for each body part. From these local axes, we can compute the orientation and the dimension of each body part (by projecting the body part's voxel on each local axis and finding the range). With the assumption that each limb consists of 2 equal segments, all joint positions can also be found.

Hand model initialization:

- For finger registration, we first compute the center of each segmented voxel region (marked with plus sign). By constructing lines from palm center C_{palm}, which we have already known (as mentioned in 3.1.2), we only use LE-based method to segment 5 fingers, and the remaining voxels are considered being of palm). Compared to all other centers, we see that the angle characteristic of the line from C_{palm} to C_{thumb} is distinguishable and can be used to register which voxel region is thumb (i.e., the minimum angle to all other lines is the largest). Other voxel regions can then be consequently registered to the remaining fingers.

- The local z-axis of each finger is computed as the largest PCA component of corresponding voxel region (marked with arrows). Similar to the body case, we use the found local z-axis to compute local x-axis and y-axis for each finger and then compute the orientation and the dimension of each finger. With the assumption that each finger consists of 3 equal segments, all joint positions can also be found (e.g., red circles).

- For the palm, the local palm z-axis is determined by the line from C_{palm} to the "lowest" joint of the middle finger. Other palm parameters are then computed similarly as described above. The result of body/hand model initialization is shown in Figure 3.2.(b) - (d).

After initialization, the KC-GMM method is used to continue inferring body/hand pose in subsequent frames.

KC-GMM Method for Pose Inference in Subsequent Frames

The implementation of KC-GMM method (step (d) in Figure 3.1) came from the previous work in [17]. The used hand model and body model are shown in Figure 2.2 (top right). For hand, there are 16 components with 27 DOF (degree of freedom). For body, there are 11 components with 19 DOF. The pose estimation procedure of this method uses the same paradigm of probabilistic clustering. Each body component is described by a Gaussian (μ_i, Σ_i) where the mean μ_i is the position and the covariance matrix $\Sigma_i = R_i \Lambda_i R_i^T$ contains information about the dimension Λ_i and orientation R_i. The set of such Gaussian components are kinematically constrained according to the predefined model. The goal is then to estimate optimal value for the Gaussian Mixture Model (GMM) under those kinematic constraints.

These kinematic constraints can be formulated by 3 equations corresponding to the three types of constraint: spherical (3 DOF) constraint, hardy-spicer (2 DOF) constraint, and revolute (1 DOF) constraint.

$$c_s(\Theta) = \mu_i + R_{0i} a_{ij} - (\mu_j + R_{0j} a_{ji}) \tag{3.1}$$

$$c_h(\Theta) = R_{0i} q_{ij} \times R_{0j} q_{ji} \tag{3.2}$$

$$c_r(\Theta) = R_{0i} q_{ij} - R_{0j} q_{ji} \tag{3.3}$$

where Θ is the embodiment of the kinematic constraints and all configuration parameters, μ_i, μ_j are the means of components i and j, R_{0i}, R_{0j} are the rotation of the components relative to the world coordinates, a_{ij}, a_{ji} are the joint positions in component coordinate frame (the origin is at the component center), q_{ij}, q_{ji} are the rotation axes of each component in either component coordinate frame. We can interpret these equations as follows: $c_s = 0$ means two joints on two component are coincided, without other constraints we have a 3 DOF joint; $c_h = 0$ means 2 rotation axes are perpendicular, combined with $c_s = 0$ we have a 2 DOF joint; $c_r = 0$ means 2 rotation axes are aligned, combined with $c_s = 0$ we have a 1 DOF joint.

These kinematic constraints are incorporated into the probability model in the form of a prior probability to have a Kinematically Constrained Gaussian Mixture Model (KC-GMM)

$$P(Y, c \mid \Theta) = P(c \mid \Theta) \prod_n P(y_n \mid c, \Theta) = P(c \mid \Theta) \prod_n \Big[\sum P(y_n \mid z_n, \Theta) P(z_n) \Big] \tag{3.4}$$

where $Y = y_n$ represents the distribution of input voxel data (generated by a mixture of Gaussians) and z_n is the hidden membership variable. The EM algorithm for this new likelihood function can then be derived for maximum likelihood estimation of Gaussian component parameters Θ, which can be interpreted to the body pose configuration.

Compared to the previous work [58], where these constraint equations are satisfied by adding a constraining step (C-step) into EM algorithm, KC-GMM method helps to provide a

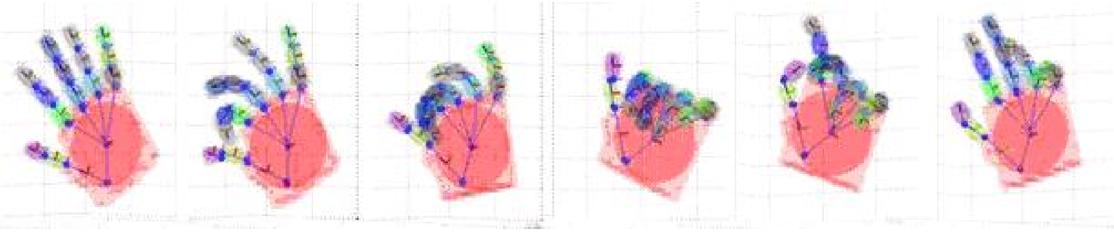


Figure 3.3: Visual results of hand modeling and tracking with synthesized hand data using the combined method.

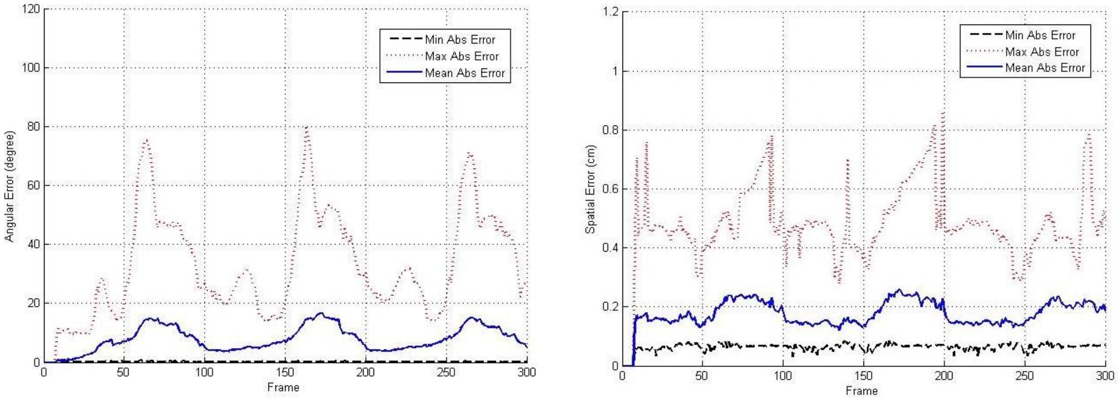


Figure 3.4: Quantitative result of synthesized hand data: Left - Angular error, Right - Position error. At a frame, errors are computed for each body component, and the min, max, mean errors are in the context of body component error.

more stable solution by removing this additional C-step (which may compete with the M-step in EM algorithm and cause instability in the optimization).

3.1.3 Experimental Results and Evaluation

The template model for body and hand (defining the number of components, the number of joint, and the number of DOF) is the same as described in [17]. For body, we experiment with real body voxel data acquired from four color cameras on the ceiling. For hand, we experiment with both synthesized hand voxel and real hand voxel. The synthesized data is constructed using the same hand model shown in Figure 2.2 (top right), and it simulates a wave pattern motion. Real hand voxel is acquired from three calibrated thermal cameras on the front screen. The background subtraction with thermal images is simple with an upper and a lower threshold respective to the temperature range of skin. Real hand voxel is then reconstructed using shape-from-silhouette technique.

The visual results of automated body/hand model initialization and tracking are quite

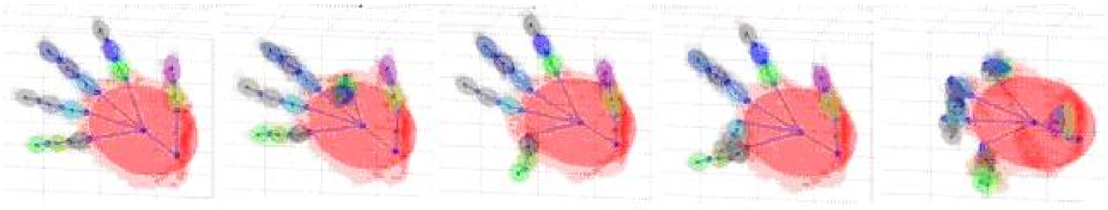


Figure 3.5: Visual results of hand modeling and tracking with captured hand data using the combined method.

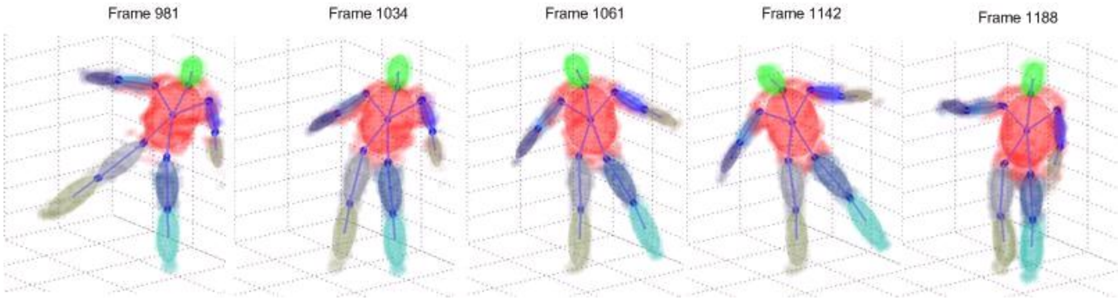


Figure 3.6: Visual results of body modeling and tracking with a sequence in HumanEva-II dataset using the combined method.

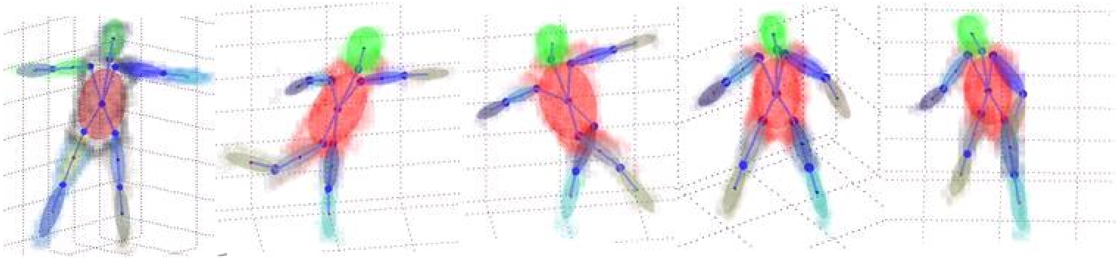


Figure 3.7: Visual results of body modeling and tracking with captured body data using the combined method.

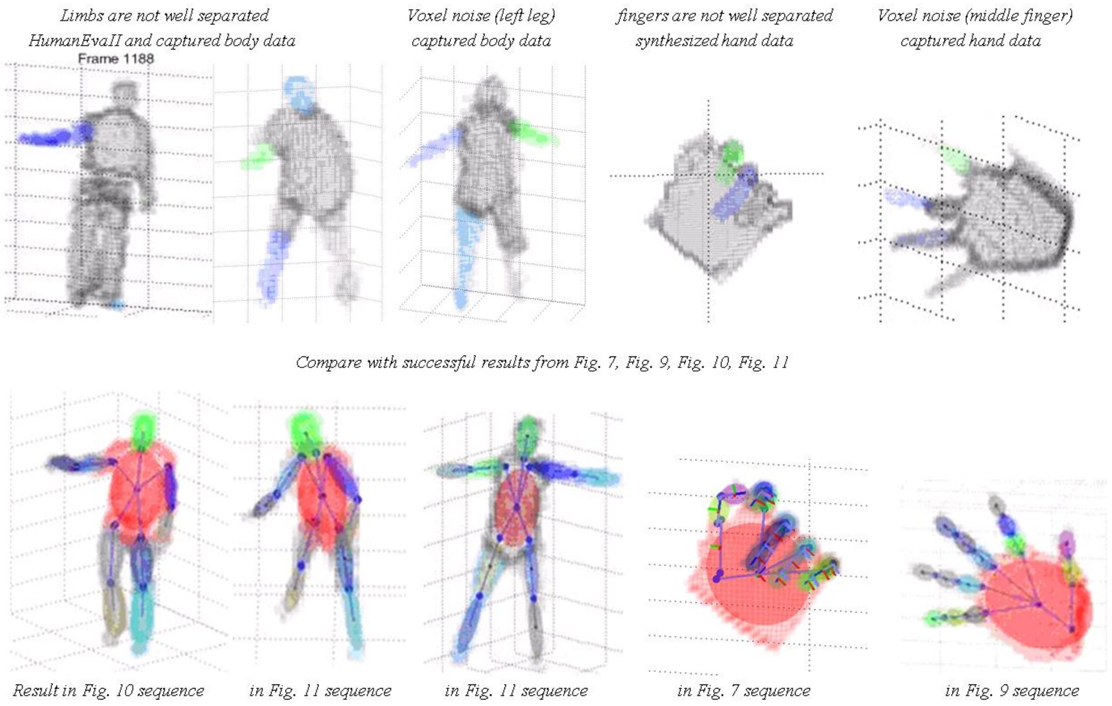
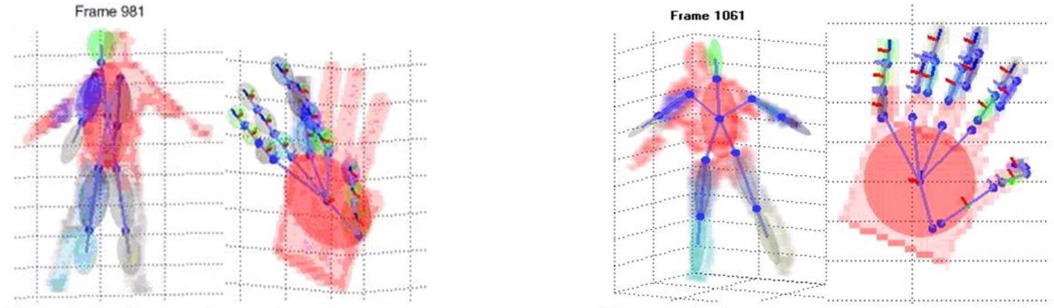

Limbs are not well separated
HumanEvaII and captured body data
Frame 1188
Voxel noise (left leg)
captured body data
fingers are not well separated
synthesized hand data
Voxel noise (middle finger)
captured hand data
Compare with successful results from Fig. 7, Fig. 9, Fig. 10, Fig. 11
Result in Fig. 10 sequence
in Fig. 11 sequence
in Fig. 11 sequence
in Fig. 7 sequence
in Fig. 9 sequence

Figure 3.8: Some failure cases of LE-based method [134] for comparison with successful tracking results of the combined method. LE-based voxel segmentation fails when limbs/fingers are not well separated and voxel noise can lead to incomplete or ghosted limbs/fingers (voxel noise is quite common in our captured data especially with the hand case).



(a) Initialized with correct dimension but incorrect orientation (rotate global orientation by 90° around z-axis)

(b) Initialized with correct orientation but with thin limbs/fingers (reduce the thickness of limbs by 50%)

Figure 3.9: Some failure cases of KC-GMM method [17] when the manual initialization is not done carefully for comparison with successful tracking of the combined method with HumanEva-II body data (Figure 3.6) and synthesized hand data (Figure 3.3). (a) KC-GMM method failed when the body/hand models are initialized with correct scale but incorrect orientation. (b) KC-GMM tracking lost quickly when body/hand models are initialized with correct orientation but incorrect size (thin limbs/fingers).

good as shown in Figure 3.3, Figure 3.5 for hand and Figure 3.6 and 3.7 for body. With synthesized hand data, we also have quantitative result of angular and position error of hand components (Figure 3.4). We see that those plots are periodic with peaks at times when the hand is in nearly closed fist pose. However, the error reduces when the hand opens and we do not lose track. Figure 3.8 and Figure 3.9 show some failure cases of the LE-based method [134] and the KC-GMM method with manual initialization [17] while the proposed combined method still track successful on the same data. In Figure 3.8, the LE-based voxel segmentation fails when limbs/fingers are not well separated and voxel noise can lead to incomplete or ghosted limbs/fingers. In our experiment with real captured data, voxel noise is quite common especially with the hand case. Figure 3.9 shows that without careful manual initialization, KC-GMM methods will fail, e.g., fail when body/hand models are initialized with correct scale (dimension) but incorrect orientation or vice versa, with correct orientation but incorrect dimension (thin limbs/fingers). This incorrectness is conceivable because of the nature of EM algorithm, which makes it easily get stuck in some suboptimal solution. We meet the same issue when there is a large displacement between frames. These results imply that our automated model initialization works well, and it is definitely a better replacement of the previous manual initialization step.

3.2 XMOB (eXtremity Movement OBservation) Framework for Real-time Upper Body Pose Tracking in 3-D

The integrated framework proposed in Section 3.1, however, was not yet implemented to be used in interactive applications. Some limitations include the computational cost related to EM algorithm (run-time performance) as well as the possibility of being stuck in suboptimal solution and cannot recover. In this section, we develop a novel system for real-time upper body pose tracking in 3-D which is suitable for interactive applications.

We know that human gestures can be seen at different levels of resolution from coarse level of full body gestures (e.g., walking, running, jumping, bending gestures [54, 159, 167], or multiperson interaction [123]) to upper-body level (e.g., waving, punching, or using driver sitting pose in predicting turn intent [16]), to more detailed levels of hand and facial gestures (e.g., hand gesture interacting with objects [106, 155], sign language [40, 69, 154], face pose tracking [55], facial action coding [7], or hand pose and face recognition with the same framework [73]). We choose to deal with the upper body part only since it is simpler than full body, less detailed than facial or hand gesture, yet conveys important information about several human activities where arms carry the most influential information of upper body motion (e.g., in meeting room, teleconference, driver assistance situations).

Motivated from studies in neurophysiology which found that the desired position of the hand roughly determines the arm posture [130], we propose a computational approach for upper body tracking using the 3-D movement of extremities (head and hands) from multiview input, called XMOB (eXtremity Movement OBservation) upper body pose tracker (a demonstration of XMOB was shown at [140]). XMOB solves the body pose estimation problem in two parts. The 3-D movements of head and hands are first tracked using multiview input. Then using upper body model constraints, the full upper body motion is inferred based on just the extremity movements as an inverse kinematics problem. The advantages are as follows: First, the challenge of an exponentially large search problem is reduced considerably by breaking it into two subproblems. Second, the self-occlusion issue is alleviated not only because of using multiview input but also because the extremities are the easier parts to track and are rarely occluded even with only two views. Furthermore, XMOB will work as long as the head and hands are observable. It does not matter if the user wears very loose clothes or the clothes colors are mixed with the background which could be a difficult case for other approaches. The downside is that we need to deal with possible pose ambiguities due to the kinematics redundancy of human body model (i.e., there could be different body poses with the same positions of head and hands).

3.2.1 Related Research Studies

In human pose estimation and tracking, we can loosely categorize related research studies into monocular [13, 25, 35, 75, 88] and multiview approaches [17, 19, 89, 136]. Compared to monocular view, multiview data can help to reduce the self-occlusion issue and provide more information to make the pose estimation task easier as well as to improve the accuracy. Since estimating the real body pose in 3-D is desirable, using voxel data reconstructed from multiview input can help to avoid the repeated projection of 3-D body model onto the image planes for comparison and the image scale issue. These advantages allow the design of simpler algorithms using human knowledge about shapes and sizes of body parts [89]. XMOB also follows a model-based approach using two-view video input and aim to extract a real 3-D posture. As concluded in [126], although human tracking is considered mostly solved in constrained situations, i.e., has a large number of calibrated cameras (> 10), people wearing tight clothes, and a static environment, there are still remaining key challenges including tracking with fewer cameras (< 4), dealing with complex environments and variations in object appearance (e.g., general clothes, hair, etc.), adapting to different body shapes with automatic initialization, and automatically recovering from failure. Based on this, Table 2.2 is a summary comparing to some extent the proposed XMOB system with selected representative model-based methods for human body pose estimation using multiview data.

Due to the exponentially large search problem of human body pose estimation and tracking, a common task is to figure out a way to reduce the search space and to search “smartly” for the optimal pose from given image evidences. Several approaches, e.g., [11, 77, 13, 88], assume some prior models of motion and/or appearance to deal with the ambiguities due to the loss of depth information. The performance of those approaches, however, is limited to the type of motion and appearance in their training data. Mikic et al. [89] use specific information about the shape and size of the head and torso to have a hierarchical growing procedure (detecting head first, then torso, then limbs) for body model acquisition and tracking. Bernier et al. [10] use a graphical model to decompose the full 3-D pose state space into individual limb state space, then use a nonparametric Belief Propagation for articulated pose tracking. In our XMOB system, we also try to reduce the complexity by breaking the large search problem of upper body tracking into two steps. The motivation came from research studies in psychophysiology [64] which showed that to recognize gesture and activity, humans do not need to observe the whole body movement but only observe the movement of some "light points" attached to the body. Later on Soechting and Flanders, researchers in neurophysiology, studied the inverse kinematics of arms and also found that the desired position of the hand roughly determines the arm posture [130]. They developed the Sensorimotor Transformation Model (STM), which is a set of linear functions, to compute angle parameters of arm pose from known end points. Kogay et al. [115] used this STM to implement an inverse kinematics algorithm for computer animation of human arms. Nevertheless, since human arms kinematics is redundant, the STM only provides one among many available solutions. To overcome this ambiguity, XMOB exploits the "temporal inverse kinematics" using observation of extremities dynamics for 3-D upper body pose tracking instead of just inverse kinematics constraints at a single frame. To some extent, XMOB tries to address some remaining key challenges as concluded in a recent summary of "state of the art" methods for pose and motion estimation including dealing with general loose clothing, recovering from failure, and using only two cameras [126].

3.2.2 Details of XMOB Framework for 3-D Upper Body Pose Tracking

The flowchart of the proposed system is shown in Figure 3.10. From two view input, XMOB first tracks head and hand blobs in 3-D based on robust semisupervised skin color segmentation. Following a numerical approach, the geometrical constraints of upper body model at each frame is used to determine a set of hypotheses for possible inner joint (shoulder and elbow joints) locations from current head and hand positions. By observing extremity movements over a period of time, XMOB deals with the ambiguity of multiple hypotheses at each frame by selecting the candidate sequence that minimizes the total joint displacement. Although minimizing the total joint displacement is a heuristic assumption, our experimental results with various subjects in different environments indicated its feasibility.

XMOB uses a skeletal model for the upper body as shown in Figure 3.10. The length of body parts including shoulder line and neck line are considered fixed, which means there is only kinematic movement at the joints. There are four joints in the model: two shoulder joints, each has 3 Degree of Freedom (DOF), and two 1 DOF elbow joints. There are also physical constraints

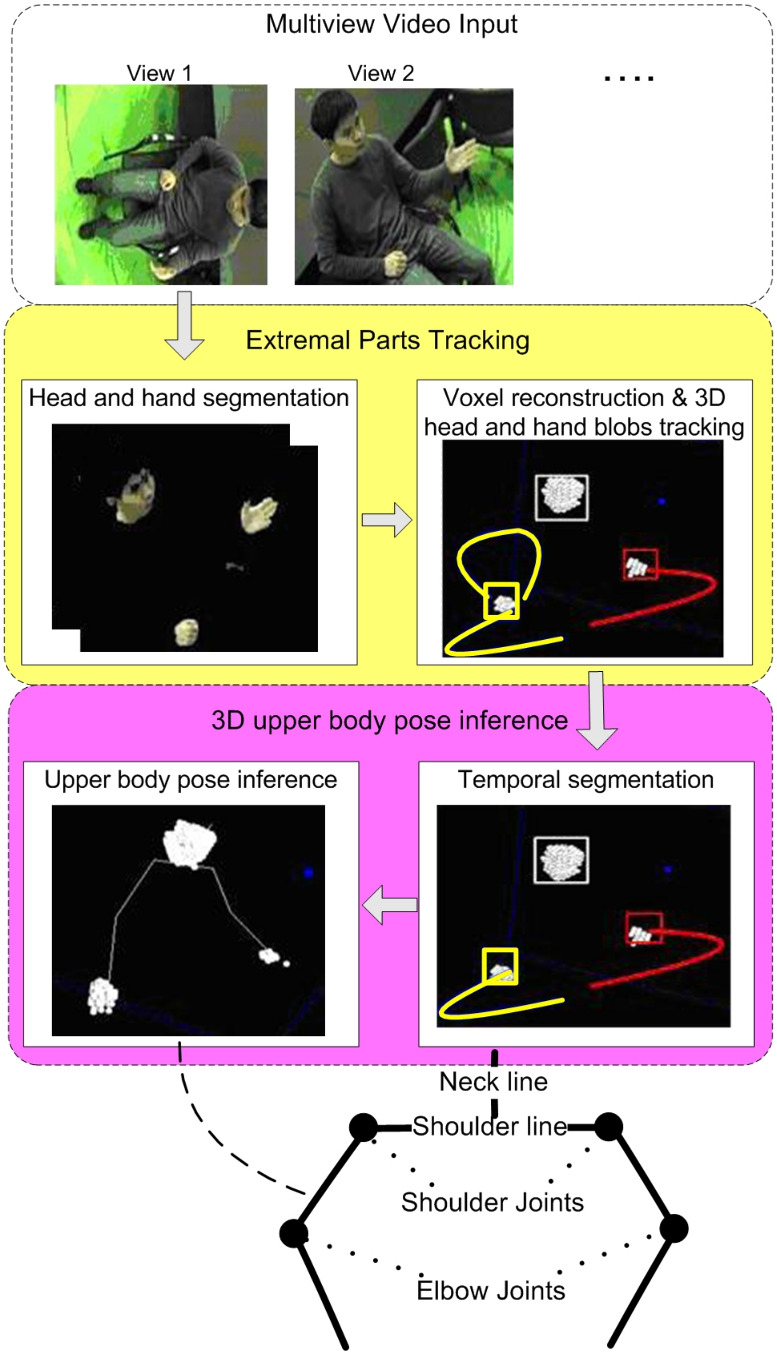

Multiview Video Input
View 1
View 2
....
Extremal Parts Tracking
Head and hand segmentation
Voxel reconstruction & 3D head and hand blobs tracking
3D upper body pose inference
Upper body pose inference
Temporal segmentation
Neck line
Shoulder line
Shoulder Joints
Elbow Joints

Figure 3.10: XMOB framework for 3-D upper body pose tracking based on head and hand movement observations. As shown, the framework uses a skeletal model of the upper body.

on shoulder joints and elbow joints which limit the possible range of joint angle within a degree of freedom. An upper body configuration can be represented by a set of upper body joint and end point positions: $Y = \{P^{hea}$,P^{lha},P^{rha}, P^{leb}, P^{reb}, P^{lsh}, $P^{rsh}\}$ which can be split into inner joints $Y^{inn} = \{P^{leb}, P^{reb}, P^{lsh}, P^{rsh}\}$ and extremal parts $Y^{ext} = \{P^{hea}$, P^{lha}, $P^{rha}\}$ in which the abbreviations are (hea: Head, lha: Left hand, rha: Right hand, lsh: Left shoulder, rsh: Right shoulder, leb: Left elbow, reb: Right elbow). We will discuss in more detail the framework components in the following sections.

Head and Hands Tracking with a Semisupervised Procedure for Robust Skin Color Segmentation

Using skin color as a cue to track head and hand blobs is quite straightforward. However, in many cases including our experimental data, merely using general skin color model, e.g., [44], is not robust enough for head and hands tracking. In principle, we will achieve more robustness and less complexity when we focus on our specific case of a particular user's skin color in a particular background compared to a general clustering model for an arbitrary user's skin color in an arbitrary background. This can be done manually if at the beginning of each session, we have a person manually select positive samples of user skin color and negative samples of background colors for that session. Our idea is to automate this process by getting some help from the interaction with user. We design a simple semisupervised procedure in which user is asked to start by trying to move only their extremities (head and hands). Combining the detected motion areas with a general skin color model, we can have a more specific and robust skin color segmentation model. From head and hand segmentation results, 3-D voxel data of head and hand blobs is reconstructed using Shape-From-Silhouette method [19]. Then 3-D head and hand blobs are tracked with mean shift algorithm.

For automatic initialization of upper body model (determination of the fixed length of the shoulder line, neck line, upper arm, and lower arm), we used a semisupervised scenario which requires the participant to start with straight arms, facing forward. With this help from the participant, the arm length can be determined and then used to scale up an average body model [87] to have the length of all parts of the skeletal upper body model.

A Numerical Method to Predict Inner Joint Sequences From Extremity Movements

Knowing the estimate of head and hand positions $\{P_t^{hea}$, P_t^{lha}, $P_t^{rha}\}$, we want to estimate the remaining inner joint positions $\{P_t^{lsh}$, P_t^{rsh}, P_t^{leb}, $P_t^{reb}\}$ as an inverse kinematics problem. Since upper body kinematics is redundant, there is no precise solution all the time, but we want to find the "normally correct" solution which could be true in many situations. XMOB uses a motion segment to perform inverse kinematics instead of single frame and add the secondary goal of minimizing inner joint displacement during that motion segment. The problem

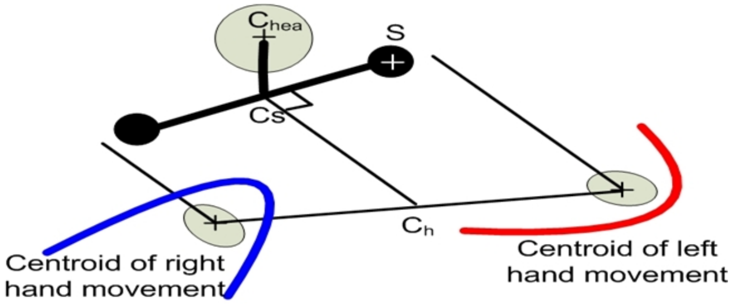

C_hea
S
Cs
C_h
Centroid of right hand movement
Centroid of left hand movement

Figure 3.11: Update the shoulder line orientation for a whole temporal segment using the constraint of preserving body left-right "balance."

is restated as follows: Given a motion segment of extremal points (head and hands) $Y^{ext}_{t1:t2}$, the goal is to find the corresponding inner joint sequence $Y^{inn}_{t1:t2}$ that satisfies the joint constraints C, and the optimization target function is to minimize the total inner joint displacement. The assumptions for XMOB can be summarized as follows:

- Most of influential information of upper body motion is carried by the arms.
- Human body has the symmetry between left and right, so during a period of time, this left-right balance tends to be preserved (although it might not be true at a single frame).
- Humans tend to optimize their movement so that the joint displacement is minimized.

Although these assumptions are kind of heuristic, our extensive experimental validation on different users with indoor and in vehicle environments implied their feasibility.

Shoulder Joint Position Prediction

Since the shoulder joints typically move with a much lower frequency than the hands and elbow joints, XMOB updates temporal shoulder position less frequently. For a given temporal segment, XMOB only predicts a single position for shoulder joints. Since human body shows a bilateral symmetry between left and right, this left-right "balance" tends to be preserved during a period of time (although it might not be true at a single frame). Note that this assumption does not mean a strict symmetry between left and right hands. As shown in Figure 3.11, this left-right "balance" assumption can be interpreted as follows: if we compute the centroid of left hand trajectory and the centroid of right hand trajectory during a temporal segment, they should be symmetric (over C_s) when projected onto the shoulder line. This also means that the shoulder line should be perpendicular to the line from the center C_h to the center C_s. From the centroid of head trajectory C_{hea}, C_s can be computed since the neck line has fixed length and is vertical. Denote a point in 3-D as a column vector of 3 coordinate and the shoulder joint is $S = [x, y, z]^T$, we have the following equations:

- Shoulder length a_1 is known from the initialization (Sect.3.2.2):

$$(S - C_s)^T (S - C_s) = a_1^2 \tag{3.5}$$

- Shoulder line is perpendicular to the neck line:

$$(S - C_s)^T (C_{hea} - C_s) = 0 \tag{3.6}$$

- Shoulder line is perpendicular to $C_h C_s$ (left-right "balance" assumption over a period of time):

$$(S - C_s)^T (C_h - C_s) = 0 \tag{3.7}$$

  where C_h can be computed as the middle of the centroids C_{lha}, C_{rha} of hands trajectories projected on the same horizontal plane at shoulder line.

Using (3.5), (3.6), (3.7), we can solve for the 3-D coordinates of shoulder joint. Equation (3.5) is quadric, so normally, we have 2 solutions S_{left}, S_{right} corresponding to left and right shoulder joints. The left and right is selected so that left shoulder is on the side of left hand and right shoulder is on the side of right hand: $(S_{left} - S_{right})^T (C_{lha} - C_{rha}) > 0$ To avoid ill-conditioned situations, we do not update shoulder joint positions when C_h is the same or close to C_s. When they are close to each other the symmetry constraint (3.7) becomes too sensitive to little changes in C_h position.

Elbow Joint Sequences Prediction

Since the length of upper arm and lower arm is fixed, possible elbow joint positions with known shoulder joint position S and hand position H will lie on a circle as shown in Figure 3.12(a). Furthermore, we realize that in a natural and comfortable position, elbow joint would lie roughly on the lower outside (points away from the body) quarter part of the circle (the bold part P_1P_2 of the circle). This can be considered as a geometric interpretation of the physical constraints on shoulder joints and elbow joints, which limit the possible range of joint angle within a degree of freedom, into an approximate geometrical constraint.

Figure 3.12(a) shows a special case when the line from shoulder joint to hand position SH is the same as y-axis. Here we assume a coordinate system attached to the shoulder joint with x-axis as the horizontal direction of shoulder line, z-axis as the vertical direction (e.g., neck line), and y-axis as the direction facing forward. Determining and quantizing the arc P_1P_2 can be done as follows. Knowing S, and H positions, the length of upper arm and lower arm, we can compute the center $C = [x_C, y_C, z_C]^T$ and the radius r of the concerned circle.

- The equations for points $P = [x_P, y_P, z_P]^T$ on the mentioned 3-D circle are:

$$y_P = y_C \tag{3.8}$$

$$(x_P - x_C)^2 + (z_P - z_C)^2 = r^2 \tag{3.9}$$

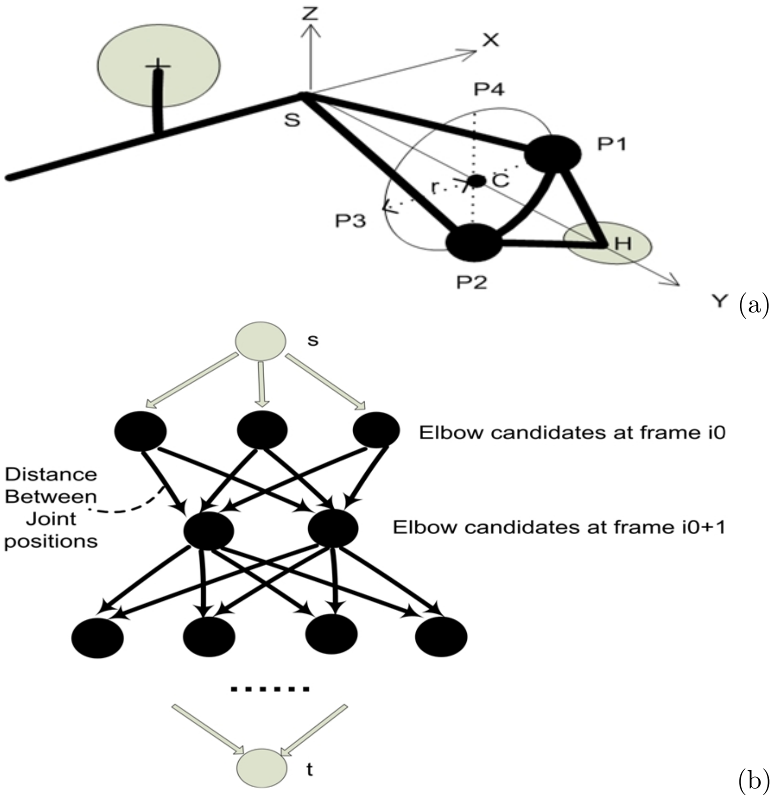


Figure 3.12: Elbow joints prediction. (a) - Generate elbow candidates at each frames. (b)-Over a temporal segment, select the sequence of elbow joints that minimizes the joint displacement. By adding 2 pseudo nodes s and t with zero-weighted edges, this can be represented as a shortest path problem.

- The outside lower quarter part P_1P_2 is determined by $z_P \in [-r, 0]$ and $x_P \in [0, r]$ for left elbow (or $x_P \in [-r, 0]$ for right elbow)

The quantizing process is done by sampling the x_P or z_P coordinate in the above range. In a general case when the hand and shoulder are in arbitrary positions, the above set of equations for a 3-D circle and the quantizing process become a bit more complex to determine and solve. We deal with this by first finding the rotation matrix R to rotate the line from shoulder joint to hand SH to y-axis to come back to the special case in Figure 3.12(a). After computing "candidates" for elbow joints in this special case, we use the inverse rotation matrix R^{-1} to transform these "candidates" back to actual elbow candidates.

The selection of elbow candidate sequence (over a temporal segment) that minimize the total joint displacement can be represented as a shortest path problem (Figure 3.12(b)). Due to the layer structure of the graph in this case, the shortest paths from the source node s to nodes in a layer are known when we reach that layer. If we save that "shortest path" to each node at

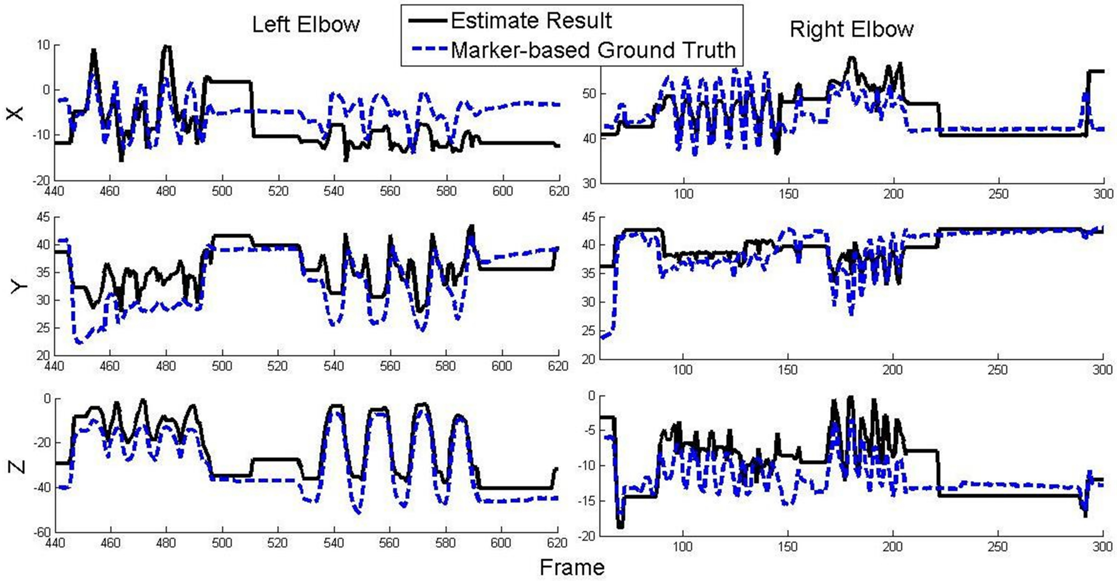


Figure 3.13: Plot of X, Y, Z coordinates of the estimated elbow position (solid black lines) compared to the ground truth (dotted blue lines) from motion capture system. We see that some movement patterns of the elbows are captured in the estimates.

current layer, the shortest paths to nodes in the next layer can be computed based on this saved information in a constant time (dynamic programming approach). Therefore, this shortest path problem can be solved in linear time complexity O(n), where n is the number of frames in the temporal segment.

3.2.3 Experimental Results and Evaluation

Upper Body Pose Tracking Results

In our experiment setup, two color cameras are configured with a wide baseline to observe the 3-D movement of head and hands. We captured several data sequences with different users in different indoor backgrounds to evaluate XMOB upper body tracking. XMOB has also been used for upper body pose tracking in vehicle environment [139]. In order to have a quantitative evaluation for some indoor sequences, we also use a marker-based motion capture system to obtain a baseline ground truth of upper body motion simultaneously with the video data for XMOB input. The system run time is about 15 fps (frame per second) on an Intel Core i7 3.0 GHz.

Table 3.2 shows the tracking error of head, left hand, right hand, left elbow, right elbow compared to the ground truth on a 2-minute sequence in which the subject did several typical gestures including clapping, waving, pushing, pointing, and some gesticulation with one hand.

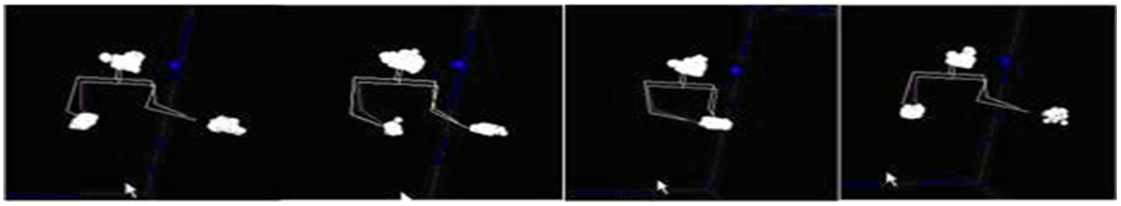


Figure 3.14: Visual results of 3-D upper body pose tracking shown with color lines. White lines are ground truth pose. White clouds are voxel data.

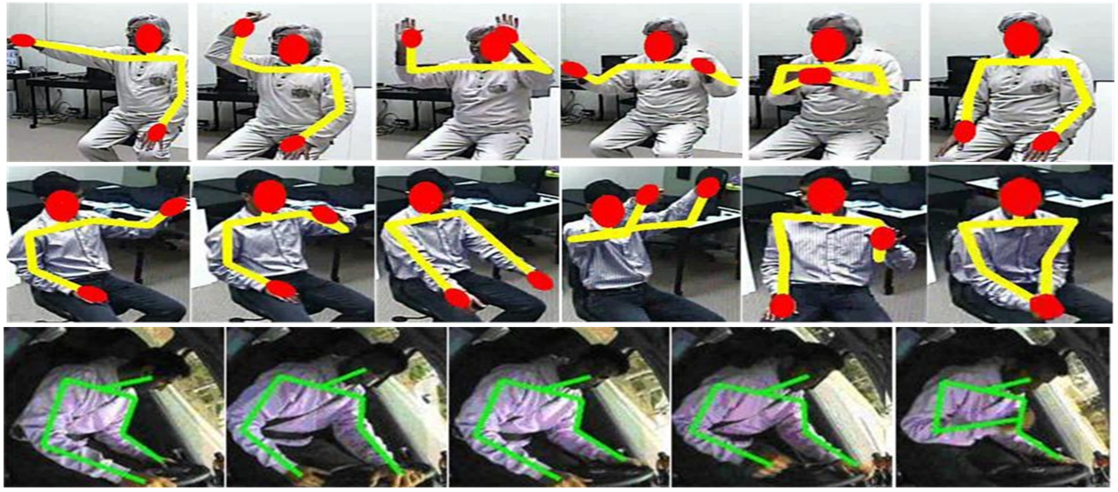


Figure 3.15: Superimposed 3-D pose tracking results on image for visual evaluation. Row 1, 2 - Sample results on a meeting room scene with different test subjects. Row 3 - Sample results on a driving scene.

Table 3.1: Spatial tracking errors on HumanEva-I boxing and waving sequences of XMOB and KC-GMM methods

	3-D Spatial Tracking Error (cm) compared to marker-based ground truth					Run time (frame per second)
	Head	LShoulder	RShoulder	LElbow	RElbow	
Boxing, KC-GMM (3 views)	11.2	15.9	18.6	52.4 (loss of track)	46.2 (loss of track)	∼ 0.1 (Matlab)
Boxing, XMOB (2 views)	6.7	10.8	12.5	12.3	14.3	∼ 15 (VC++)
Waving, KC-GGM (3 views)	4	4.4	5.7	16.6	27.1 (loss of track)	∼ 0.2 (Matlab)
Waving, XMOB (2 views)	5.5	5.6	6.2	6.4	8.2	∼ 15 (VC++)

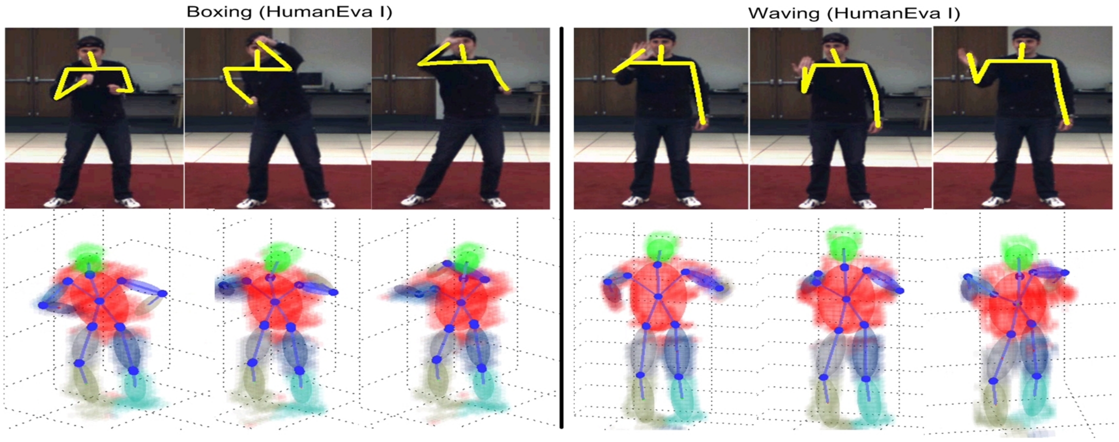


Figure 3.16: Sample results on the boxing (left) and waving (right) sequences from the public HumanEva-I dataset. Top row: Superimposed results from XMOB (using 2 views). Bottom row: Results from Kinematically Constrained Gaussian Mixture Model method using 3 views.

Table 3.2: Spatial tracking errors compared with the ground truth. Since the ground truths here are not the exact joint positions, there are some base errors in these tracking results (see section 3.2.3 for more details)

	3-D Tracking Error (cm)	
	Mean	Variance
Head	5	1.2
Left Hand	11.9	7.0
Right hand	10.7	2.3
Left Elbow	7.8	5.0
Right Elbow	6.5	5.3

Note that the ground truths in this experiment are also not the exact joint positions, e.g., to avoid occlusion of markers on some skin regions of hand, we put markers on the wrist to detect hand position. Therefore, the mean error might be quite large due to this base error. Because the real error can be an addition or subtraction from this base error, we cannot simply subtract this base error from the measured error. However, the variance of the measured error will give us a sense of the order of the real error. Figure 3.13, which shows directly the estimated 3-D position of elbows compared to the ground truth, indicates that these estimates can capture the movement pattern of joints. We know that even the same person has variance in doing the same gesture (e.g., in clapping gesture, people can clap in different direction, at different height) and different people will have more variance in doing the same gesture. Therefore, with regard to gesture analysis, the movement patterns provide more influential information than the exact joint positions.

Some results for visual evaluation are shown in Figure 3.14 (the visual result of 3-D upper body pose tracking compared to the ground truth). Figure 3.15 shows visual evaluation of the pose tracking result superimposed on input image with different subjects in both indoor and in vehicle environments.

Comparison with Kinematically Constrained Gaussian Mixture Model (KC-GMM) Method on The Boxing and Waving Sequences from The Public HumanEva-I Dataset

We apply XMOB on two color views of the boxing and waving sequences in HumanEva-I dataset. For an indicative comparison, we also apply the KC-GMM method [17] on all three color views of these sequences. Figure 3.16 and Table 3.1 show the qualitative and quantitative comparison regarding the spatial tracking accuracies (with marker-based ground truths from HumanEva-I) and the run time. We see that since the KC-GMM method needs to use the voxel data of the full body, it has an issue when there are noise and missing voxel data due to noise in image segmentation and limited number of camera views. Actually, the KC-GMM method loses track of the arms just a while after the manual initialization of the start pose which resulted in high tracking errors of the elbows. On the other hand, XMOB only needs the voxel data of head and hands so it could work pretty well with only two views. We did observe some failure cases of XMOB: When hands are too close, there is a misassignment between left and right hand (cross arms vs. normal arms). However, this misassignment was recovered when hands move apart. In Figure 3.16, the second image of XMOB results on boxing sequence indicated a situation when the assumption of left-right balance in estimating the shoulder line does not hold (equation (3.7)). In such cases, we have larger errors in shoulder joints and elbow joints estimation. However, it is important that XMOB can still work and will recover when these difficulties disappear. With regard to the run time, although we need to take into account that KC-GMM ran in Matlab (on the same machine), the speed difference is still considerable. Note that the run time of KC-GMM also varies considerably depending on how many iterations are needed to converge to an acceptable GMM fitting (e.g., KC-GMM will take a longer time to run when the movement displacement is larger).

3.3 Combining Human Pose Estimation and Tracking at Multiple Levels

As discussed in Section 3.1, achieving multilevel description of a full body model (e.g., including body, hand, and head) is useful and desirable. However as far as we are concerned, published research studies only deal with each task of estimating body pose, hand pose, or head pose separately. There are some reasons for this fact. First, the mentioned high-dimensional issue becomes more serious if we want to deal with full body model estimation, e.g., for a full model of body with 2 hands, we have 19 DOF (of the body) + 2*27 (of two hands) DOF = 73 DOF. We may say that this is an explosion in pose configuration space, so applying current methods to solve this huge problem in one shot will be very inefficient or even impossible. Second, the difference in size between body and hand leads to difficulty in achieving good data (e.g., voxel data) as well as estimating the pose of both body and hand in one shot. In our proposed framework (Figure 3.17), we do not try to tackle these difficulties directly but try to find a “roundabout” way to reach the final goal of a full body model: We still use different camera arrays for body, hand and head to be able to have good data for each task. The huge problem of estimating full model of human body is still broken into different tasks of estimating body pose, hand pose and head pose. These tasks are done separately therefore the issue of search space exploding will not arise. Different camera arrays are calibrated into the same extrinsic world coordinates and capture input data for body, hand and head simultaneously so that, at the final step, we can have a module to combine the achieved results of each task into a full model of human body. At this time, our combination module simply connect the estimated results of body, hand, and head into a full model using calibration parameters. Estimating a full model including body, hand, and head is a new area and this is just an initial step into it. In the future work, this combination (fusion) can happen at lower levels, e.g., at pose estimation level.

Besides the common issues in each task such as high-dimensional issue (i.e., even within each task the number of DOF is still high), occlusion issue, rapid motion issue (e.g., hand motion can be very fast), and real-time requirement issue, this framework brings out some additional issues that we need to solve. First, the setup and calibration between camera arrays that capture data at different level of resolution is more complex and we need to deal with error propagation issue in calibration using intermediate common coordinate systems. Second, because most of the methods for articulated body pose estimation from voxel data only work when the given voxel data is of the concerned object only, we need to segment the voxel data for each concerned body

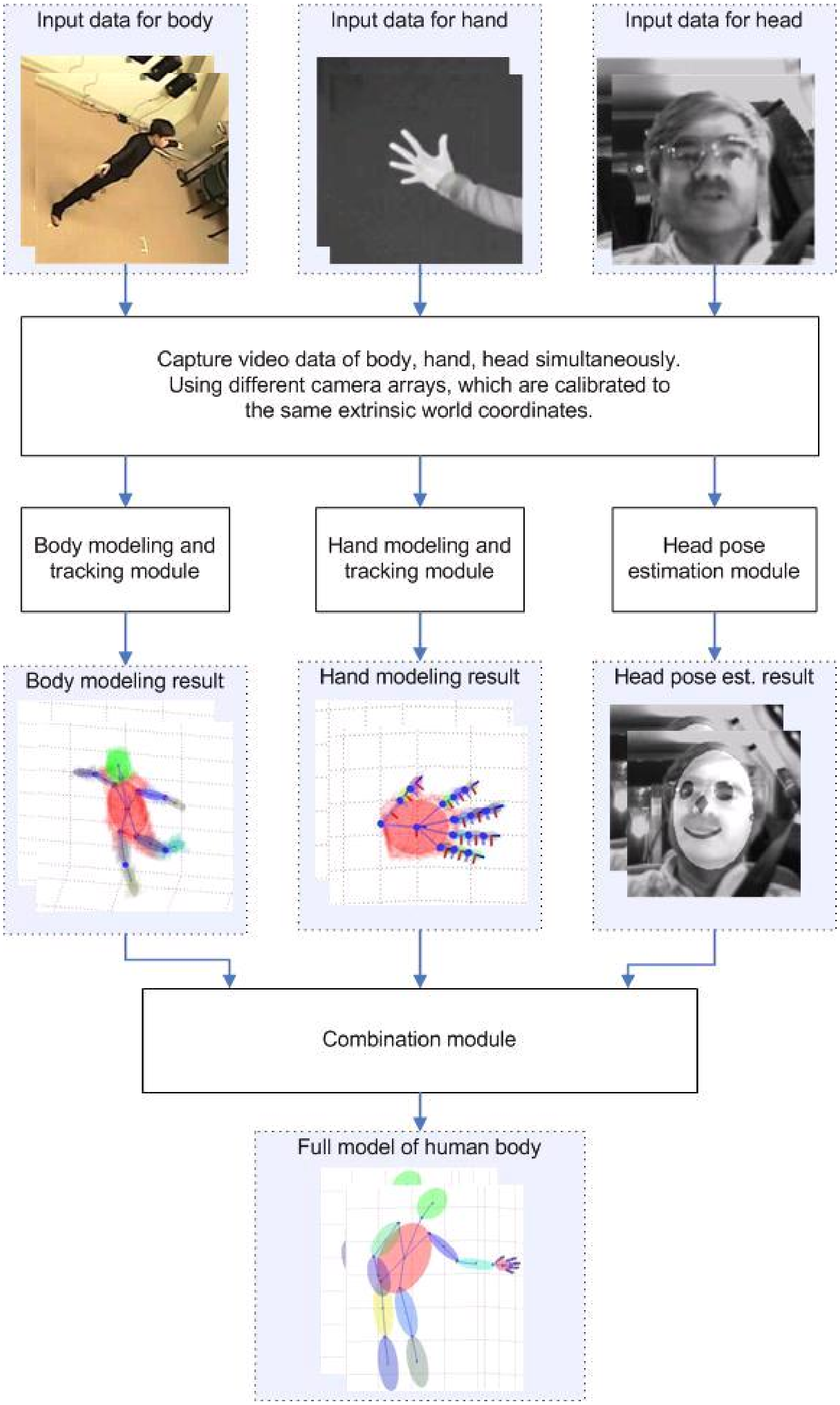


Figure 3.17: An integrated framework for human pose estimation at multilevel in order to elaborate full model of body, hand, and head.

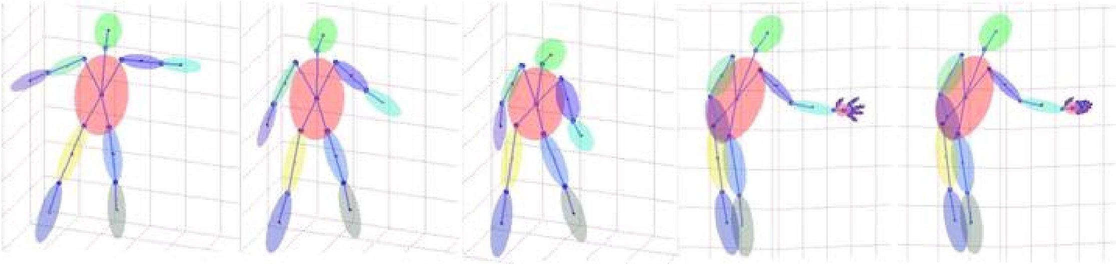


Figure 3.18: Visual results of combining achieved body model and hand model into a full model of body and hand.

part (e.g., segment hand voxel data from voxel data of lower arm). In our experiment, we tried to deal with these issues to some extent. However, these issues remain unsolved in general and require more research studies.

3.3.1 Preliminary Experiment and Results

We apply the integrated framework with automatic initialization in Section 3.1 to both body and hand modeling and tracking. The experimental setup is shown in Figure 3.19. There are 2 camera arrays: The first one consists of 4 color cameras on the ceiling to capture body data and the second one consists of 3 thermal cameras on the front wall to capture hand data. These two camera arrays are calibrated to the same extrinsic world coordinates and the data for body and hand are captured simultaneously. As discussed above, there are some issues that we have to face with when applying this integrated framework. Regarding calibration issue, we configured the two camera arrays for body and for hand so that they can see a common subregion and used Caltech Matlab calibration toolbox to calibrate the two camera arrays to the same real world coordinate system placed at the common area (Note that we may not need this common area by calibrating each camera array to its own real world coordinate system then converting between the two real world coordinate system. However, there will be issue of error propagation in doing so). This camera configuration has some limitations, i.e., when capturing hand data, the hand needs to be in some limited positions in order to reduce the ambiguity in voxel reconstruction. Regarding the issue of segmenting hand voxel data from lower arm voxel data, we use thermal cameras for hand so the background subtraction for hand region can be done easily by setting empirical upper and lower brightness thresholds respective to the temperature range of skin.

The experimental scenario is that a person comes into the room and moves around. He then goes to the front screen and put his hand into the region that thermal cameras for hand can capture data. The body modeling and tracking and hand modeling and tracking are still done separately but their results can be combined into a full model of both body and hand. Currently, our combination module simply connect the estimated results of body, hand

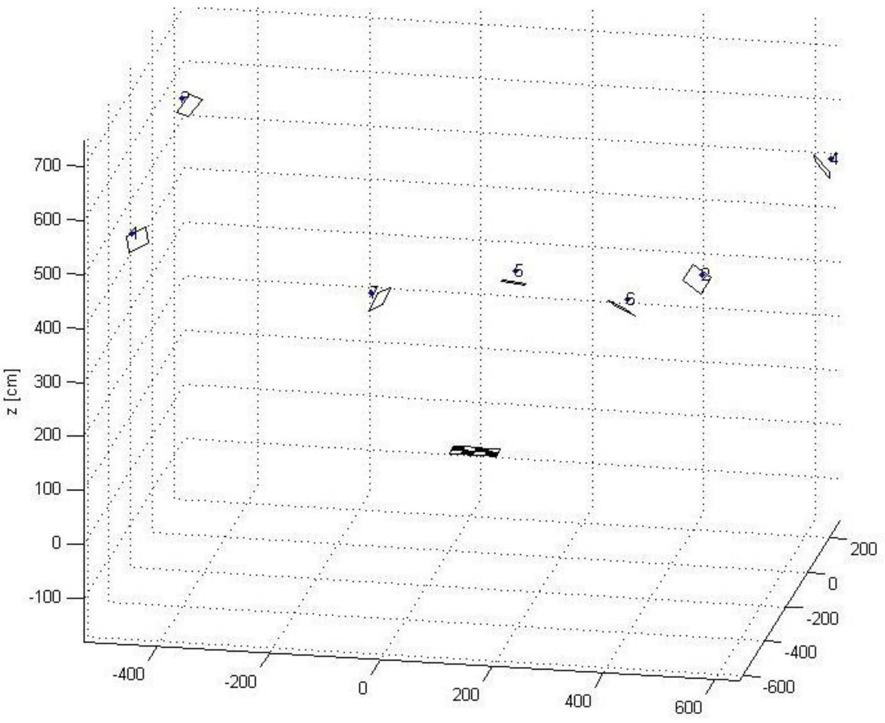


Figure 3.19: Experimental setup for observing human body and hands - 4 color cameras on the ceiling to capture body data and 3 thermal camera on the front wall to capture hand data.

into a "full" model using calibration parameters. The combination (fusion) at lower levels, e.g., at pose estimation level is leaved for future work. Because we do the experiments according to our integrated framework with calibrated camera arrays and simultaneous body/hand data capturing, the procedure to connect resulted body model and hand model into a full model is straightforward. In our experiment, this combination process works quite well although there is sometimes a small mismatch in the position of the hand model and the position of the lower arm in the body model. This could be the result of errors in body and hand modeling and tracking. In our experiment, we have done a simple adjustment by constraining the wrist joint on the lower arm and wrist joint on the hand to be the same. The combined result of "full" model of body and hand is shown in Figure 3.18. The first three images are when the person moves around in the room. The last two images are when the person goes to the front screen to capture hand data, the resulting body model and hand model are now combined into a full model.

3.4 Acknowledgments

Chapter 3 is based on material that is published in the International Conference on Pattern Recognition (2008) and IEEE Transactions on Industrial Informatics (2012) both by

Cuong Tran and Mohan M. Trivedi. The dissertation author was the primary investigator and author of these papers. This research is sponsored in part by U.C. Discovery Program and National Science Foundation. We would like to thank our colleagues in the Computer Vision and Robotics Research laboratory for their helpful suggestions and assistance.

Chapter 4

Multilevel Human Gesture Analysis for Interactivity

Based on the XMOB upper body tracker described in Section 3.2, we present in this chapter our development of the very first system, as far as we are concerned, that does both 3-D upper body pose tracking in real time and gesture recognition based on the pose tracking outputs (i.e., joint angle dynamics). We then introduce our driver assistance system for "keeping hands on the wheel, eyes on the road" using combined tracking information at different levels of upper body and head. With a focus on the application domain of intelligent driver assistance, we also develop an efficient framework for driver foot and head behavior analysis based on optical flow tracking.

4.1 Human Gesture Recognition Based on Pose Tracking Output

Human gesture recognition from vision input is challenging due to human variations in doing the same gesture (intraclass), e.g., differences in human appearance, viewpoint, and action execution as well as the overlap between gesture classes (interclass). Two main components of an action recognition system are choosing an action representation (feature) space and then action classification. It is very important to select a good representation space which should generalize over variations within each gesture class but still is rich enough to distinguish between different classes. Some examples of action representation space include simple global motion extracted by frame differencing [121], space-time shape, motion history volume [111], cylindrical voxel histogram [54], and distance transform of body contours [159]. Since extremal parts (i.e., head and hands in case of upper body) can be extracted more reliably with less occlusion compared to

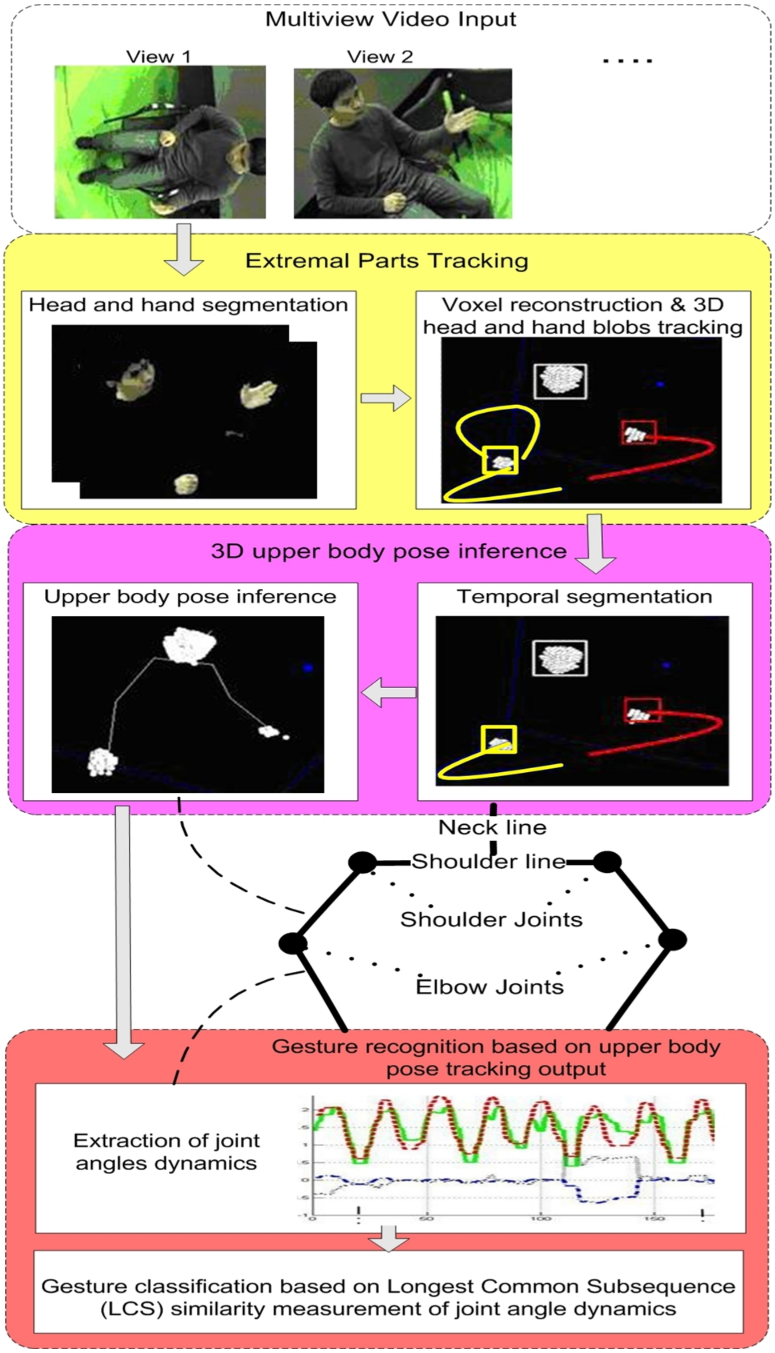

Multiview Video Input
View 1
View 2
. . . .
Extremal Parts Tracking
Head and hand segmentation
Voxel reconstruction & 3D head and hand blobs tracking
3D upper body pose inference
Upper body pose inference
Temporal segmentation
Neck line
Shoulder line
Shoulder Joints
Elbow Joints
Gesture recognition based on upper body pose tracking output
Extraction of joint angles dynamics
Gesture classification based on Longest Common Subsequence (LCS) similarity measurement of joint angle dynamics

Figure 4.1: Framework for upper body gesture recognition based on XMOB 3-D upper body pose tracker.

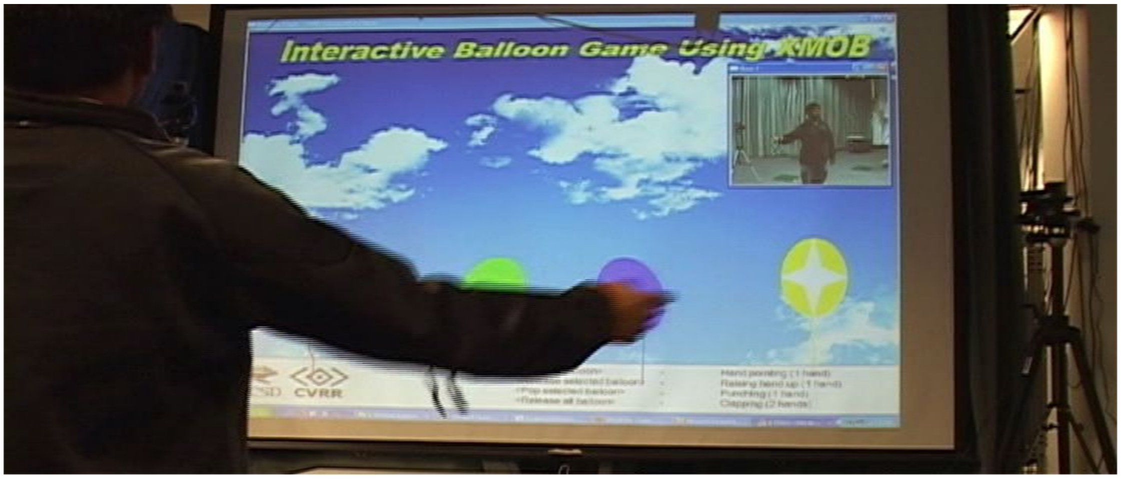


Figure 4.2: An interactive game in which user uses some upper body gestures like pointing, punching, clapping to select, pop, and release balloons

other inner parts, there are also lots of methods using features based on extremities dynamics, e.g., hand motions and posture [74, 119], head and hands trajectories [101], or variable star skeleton representation [167]. The proposed system is also based on extremities (head and hands) movements for gesture recognition. However, instead of using raw head and hands trajectories, we incorporate knowledge of the underlying upper body model which could help improve gesture recognition. An intuitive and clear way to do so is to implement upper body pose estimation and tracking. The output of body pose tracking such as joint angle dynamics are mentioned as rich, view-invariant representations for gesture recognition but challenging to derive [111].

Based on the XMOB framework, we develop the very first system, as far as we are concerned, that does both 3-D upper body pose tracking in real time and then gesture recognition based on the pose tracking outputs (Figure 4.1). Using the joint angle dynamics from 3-D upper body pose tracking, gesture classification is done based on Longest Common Subsequence (LCS) similarity measurement of joint angles dynamics. It should be mentioned that continuous body movement may contain both gesture movements and nongesture movements. Therefore the task of gesture spotting (extracting a gesture segment, which will be classified, from a continuous movement sequence) is also important and challenging. In this work, we have not actually dealt with the gesture spotting issue yet. Regarding our experiment for gesture recognition, the subject is required to perform only pre-determined gestures separated by a stop period. Therefore we can simply segment the gesture part based on the motion vs. nonmotion cue. Our experimental results for gesture recognition showed good classification rate (over 90% in average) for 6 common upper body gestures indicating the advantage and feasibility of developing gesture recognition system based on pose tracking. We have also applied this system to develop an interactive game

(Figure 4.2) in which the subject can use some common gestures to interact with the balloons (a supplemental video clip is available. [1])

4.1.1 Longest Common Subsequence (LCS) Similarity Measurement

In the used skeletal model for upper body (Sect.3.2.2), we have totally 8 joint angles (3 for each shoulder joint and 1 for each elbow joint). The joint angle dynamics in a temporal segment $\{A^{joint}\}^{joint=1:8}$ can then be extracted from upper body pose tracking output. To measure the similarity between joint angle sequences, we chose LCS measurements which has been used for trajectory similarity measurement and has been shown to be more robust to noise than Euclidean and Dynamic Time Warping [157]. Consider two sequences $A = (a_1, ..., a_m)$ and $B = (b_1, ..., b_n)$

$$LCS(A,B) = \begin{cases} \text{0 if A or B is empty} \\ \\ \text{1+LCS(Head(A),Head(B))} \\ \text{ if } |a_m - b_n| < \epsilon_1 \text{ and } |m-n| < \theta \text{ and} \\ |a_{m-1} - a_m| > \epsilon_2 \text{ and } |b_{n-1} - b_n| > \epsilon_2 \\ \\ \text{Max\{LCS(Head(A),B),LCS(A,Head(B))\}} \\ \text{otherwise} \\ \end{cases} \tag{4.1}$$

where Head(A) is the remaining sequence of A after removing the last element; control thresholds: ϵ_1 (determines if elements a_m in A and b_n in B are matched or not), ϵ_2 (measure similarity only when the joint angles are changing), θ (tolerates some time shifting in matching the two sequences). Dynamic programming is used to avoid the massive recursive computations.

Here we applied the LCS algorithm in [157] to compute the similarity between joint angle sequences with a small change to measure the similarity only when joint angles are changing (shown in equation 4.1). This helps to avoid the case when the similarity of unchanged joint angles (not useful for gesture recognition) dominates the similarity of acting joint angles.

4.1.2 Nearest Neighbor Clustering Based on LCS Measurements for Gesture Recognition

Denote $T_k = \{I_i\}_{i=1:n}$ as the training set for gesture k where n is the number of training samples. We compute the similarity between each pair of I_i, I_j in T_k

$$\begin{aligned} &S_k = \{s_{ij}\}^T_{i,j=1:n,i\neq j} \text{ where} \\ &s_{ij} = sim(I_i, I_j) = \{LCS(A^1_i, A^1_j), ..., LCS(A^8_i, A^8_j)\} \end{aligned} \tag{4.2}$$

[1] http://cvrr.ucsd.edu/ctran/Supplements/TII-SpecialIssue-Supplement.avi

Consider S_k a multivariate vector of similarity at each joint angle, we compute the mean and covariance

$$\begin{aligned} E_k &= mean(S_k) = \mu^1, \mu^2, \mu^3, \mu^4, \mu^5, \mu^6, \mu^7, \mu^8 \\ B_k &= cov(S_k) \end{aligned} \tag{4.3}$$

E_k characterizes gesture k, e.g., for RA (Right Arm) punching gesture we have a E_k pattern with high similarity in μ^5, μ^6. The centroid sample I_k for gesture k is chosen from T_k so that

$$I_k = argmin_i \sum_{j=1:n, j \neq i} D(s_{ij}, E_k) \tag{4.4}$$

Where D is the Mahalanobis distance

$$D(s, E_k) = \sqrt{(s - E_k)^T \times B_k^{-1} \times (s - E_k)} \tag{4.5}$$

Given a test sample I, the distance from I to a gesture cluster k is computed as

$$DIS_k = D(sim(I, I_k), E_k) + D(sim(I, I_k), sim(I, I)) \tag{4.6}$$

in which, the first term $D(sim(I, I_k), E_k)$ measures how test sample I is close to gesture cluster k while the second term takes into account how well the cluster k characterize the joint angles dynamics in I. For example the first term may have a high similarity score if part of the joint angle dynamics in gesture I (e.g., $A^{1,2}$) has similar motion pattern as in cluster k. However, if gesture I also has joint angle dynamics (e.g., $A^{5,6,7}$) which do not appear in cluster k, the second term will help to tell that cluster k does not fully characterize gesture I. We assume a "cluster" with the mean $sim(I, I)$ and identity covariance matrix so roughly the second term becomes the Euclidean distance from $sim(I, I_k)$ to $sim(I, I)$. The sum DIS is used to choose the closest gesture cluster.

4.1.3 Experimental Results and Evaluation

The experiment is done on a set of 4 one-arm gestures: LA punching, RA punching, LA waving, RA waving and 2 two-arm gestures: Clapping and Dumbbell curls. There are 5 subjects with different skin tone and height. Each subject performs each gesture 10 times. Two thirds of them are chosen randomly for training and the other one third is used for testing. This testing with random selection of training and testing set is repeated 5 times. The average values over different runs are then computed. Table 4.1 shows the confusion matrix (average values over 5 runs) of 6 gestures for all 5 subjects. The statistics of the recall for 6 gestures (the diagonal of the confusion matrix) is shown in Table 4.2. Some variances of $\sim 10\%$ on different runs indicate that we may need a large dataset for better evaluation.

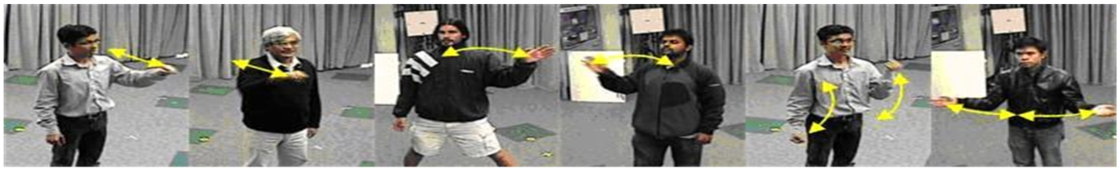


Figure 4.3: Six gestures for recognition: LA (Left Arm) punching, RA (Right Arm) punching, LA waving, RA waving, dumbbell, clapping.

Table 4.1: Accuracy confusion matrix of 6 gestures for all 5 subjects (average value over 5 runs). Each row sums up to 100%.

	LA punch	LA wave	RA punch	RA wave	Clap	Dumbbell
LA punch	**90**	*10*	*0*	*0*	*0*	*0*
LA wave	*2.4*	**97.6**	*0*	*0*	*0*	*0*
RA punch	*0*	*0*	**91**	*9*	*0*	*0*
RA wave	*0*	*0*	*2.4*	**95.2**	*2.4*	*0*
Clap	*0*	*5*	*0*	*10*	**85**	*0*
Dumbbell	*0*	*0*	*0*	*0*	*0*	**100**

Table 4.2: Statistics of the recall for 6 gestures (over 5 runs).

	Recall (in %) for 6 gestures (over 5 runs)	
	Mean	Variance
LA punching	90.0	10.0
LA waving	97.6	5.4
RA punching	91.0	11.0
RA waving	95.2	6.6
Clapping	85.0	13.7
Dumbbell	100.0	0.0

4.2 Driver Assistance System for Keeping Hands on The Wheel and Eyes on The Road

Motivated from a basic tip for safe driving, "Keeping hands on the wheel and eyes on the road," we introduce a vision-based system for driver activity analysis by observing 3-D movement of driver's head and hands from multiview video. From the results of upper body and head pose tracking, semantic descriptions of driver activities are extracted in two steps: First, we determine basic activities of each upper body part (e.g., two, one, or no hand is on the steering wheel; head is looking left, straight, or right). Then these basic activities are combined in a fusion step to extract higher level of semantic description of driver activities (e.g., whether the driver is following the above safety tip or not). Our experimental evaluation with real-world street driving shows the promise of applying the proposed system for both post analysis of captured driving data as well as for real-time driver assistance.

4.2.1 Details of The Proposed System

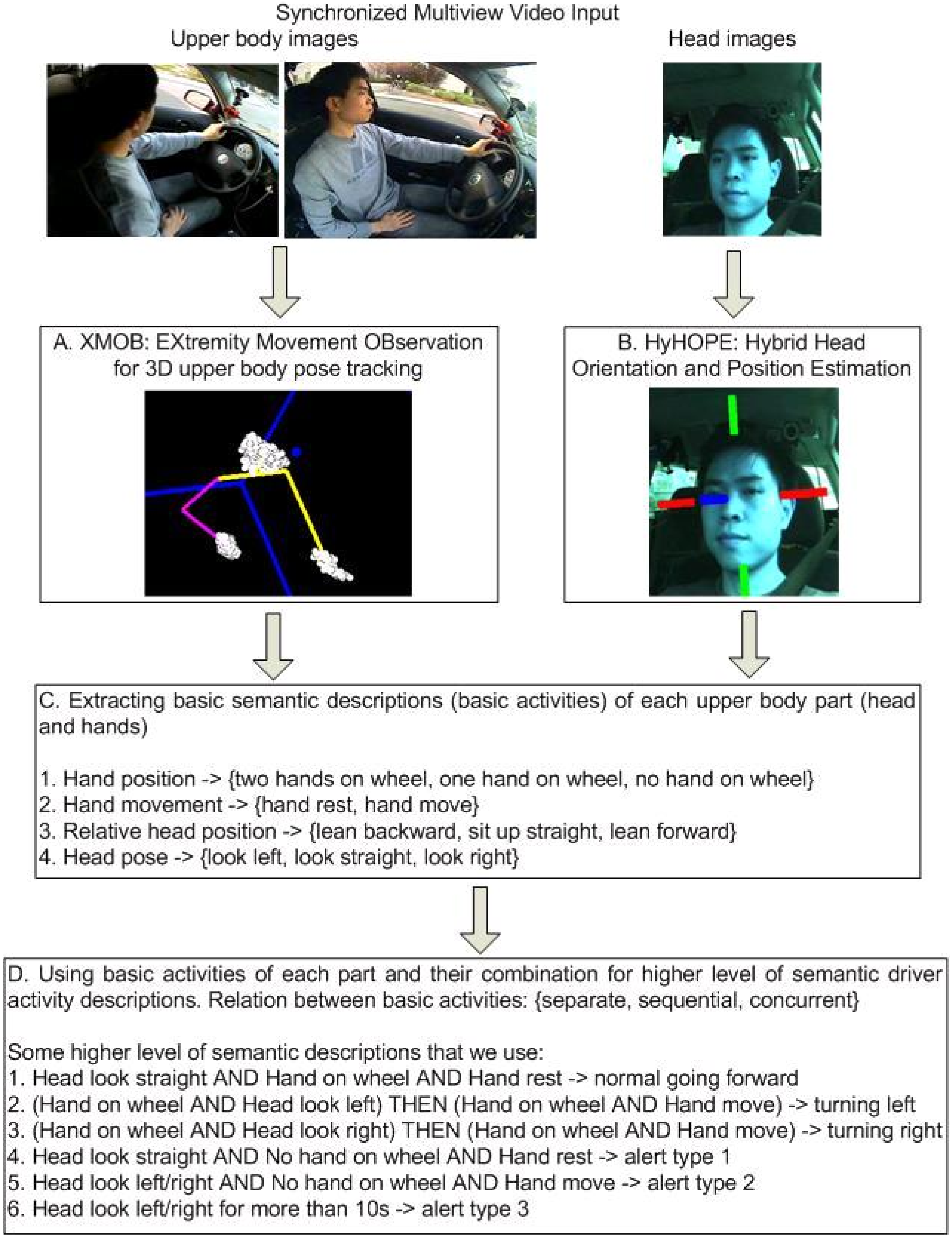
The flowchart of the proposed system is shown in Figure 4.4. From synchronized multiview video input, the upper body pose and head pose are tracked by XMOB and HyHOPE respectively in two different streams. However, the pose estimation results are then synchronized and used for extracting semantic descriptions of driver activities in two steps: First, we determine basic activities of individual upper body part such as determine if the head is looking left, straight, or right; if hand is in rest or moving state (currently we use 4 set of basic activities as shown in Figure 4.4.C). Then there is a fusion step using these basic activities as input for higher level of semantic description of driver activity. Figure 4.4.D shows the 6 types of semantic driver activity descriptions (events) that we currently detect.

HyHOPE Head Pose Tracking

We use the implementation HyHOPE in [98] for head pose tracking. The main idea of HyHOPE is to improve the performance by combining a static head pose estimation with a real time 3-D model-based tracking system. From an initial estimate of head position and orientation, the system generates a texture mapped 3-D model of the head from the most recent head image and using particle filter approach to find the best match of this 3-D model from each subsequent frame. HyHOPE also exploit GPU (Graphics Processing Unit) programming to run in real time.

Extracting Basic Activities of Each Upper Body Part

There are 4 set of basic activities as shown in Figure 4.4.C. These basic activities are simply extracted from upper body pose and head pose tracking results by a thresholding process. For set 1, we roughly mark a 3-D region for the steering wheel and determine if hand position



A. XMOB: EXtremity Movement OBservation for 3D upper body pose tracking

B. HyHOPE: Hybrid Head Orientation and Position Estimation

C. Extracting basic semantic descriptions (basic activities) of each upper body part (head and hands)

1. Hand position -> {two hands on wheel, one hand on wheel, no hand on wheel}
2. Hand movement -> {hand rest, hand move}
3. Relative head position -> {lean backward, sit up straight, lean forward}
4. Head pose -> {look left, look straight, look right}

D. Using basic activities of each part and their combination for higher level of semantic driver activity descriptions. Relation between basic activities: {separate, sequential, concurrent}

Some higher level of semantic descriptions that we use:
1. Head look straight AND Hand on wheel AND Hand rest -> normal going forward
2. (Hand on wheel AND Head look left) THEN (Hand on wheel AND Hand move) -> turning left
3. (Hand on wheel AND Head look right) THEN (Hand on wheel AND Hand move) -> turning right
4. Head look straight AND No hand on wheel AND Hand rest -> alert type 1
5. Head look left/right AND No hand on wheel AND Hand move -> alert type 2
6. Head look left/right for more than 10s -> alert type 3

Figure 4.4: Flowchart of the proposed system for driver activity analysis.

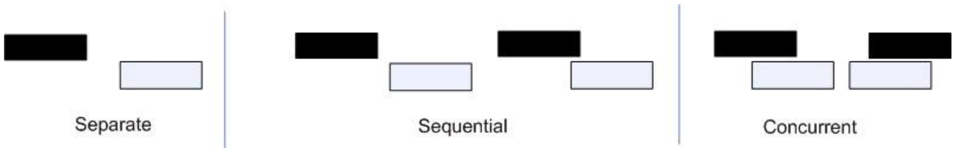


Figure 4.5: Three types of relation between basic activities.

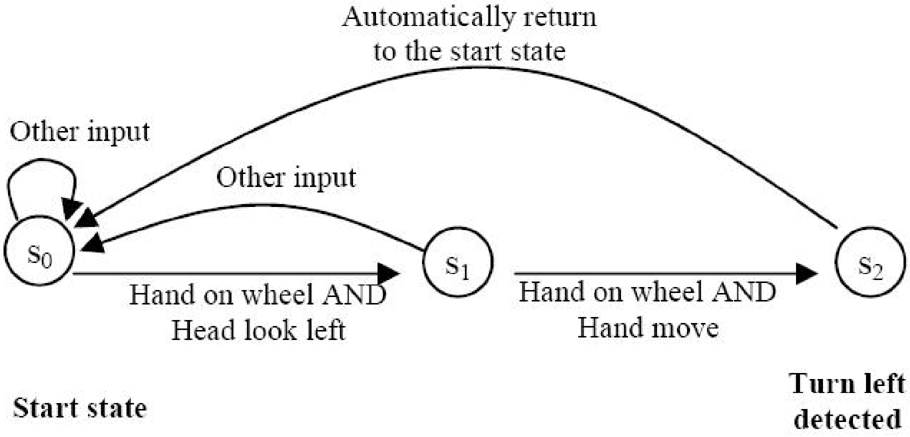

Automatically return to the start state
Other input
Other input
S_0
S_1
S_2
Hand on wheel AND Head look left
Hand on wheel AND Hand move
Start state
Turn left detected

Figure 4.6: Example of a 3-state state machine for rule 2 in Figure 4.4.D.

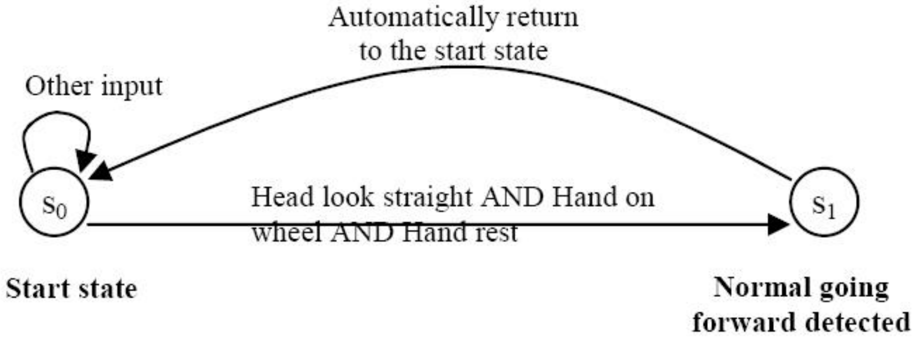

Automatically return to the start state
Other input
S_0
S_1
Head look straight AND Hand on wheel AND Hand rest
Start state
Normal going forward detected

Figure 4.7: Example of a 2-state state machine for rule 1 in Figure 4.4.D.

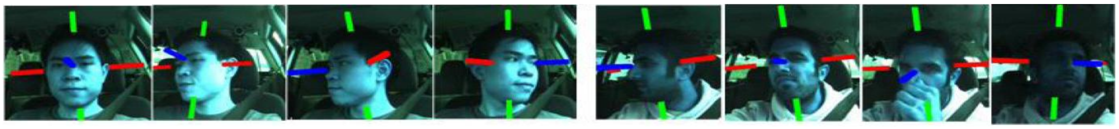


Figure 4.8: Visual evaluation of HyHOPE head pose tracking with large range of head rotation, change in lighting condition, and some occlusions.

is inside that region (hand on wheel) or not. For set 2, a distance threshold (which is 10cm in our experiment) between consecutive hand positions is used to determine if hand is in rest or motion state. For set 3, since we assume that the driver sitting in a fixed position, sitting pose is determined by a distance threshold (which is 20cm) between the referenced 3-D head position and current 3-D head position projected on the direction of car length. For set 4, an angle threshold of head pan angle (which is 200 in our experiment) is used to determine looking left/straight/right.

Combining Basic Activities of Each Upper Body Part for a Higher Level of Semantic Description

We represent the relation between basic activities as separate, sequential, or concurrent based on their time gap as well as the percentage of overlapping between them (as illustrated in Figure 4.5).

As shown in Figure 4.4.D, the fusion of basic activities into higher level of semantic description works in a rule-based manner in which AND operator represents concurrent relation and THEN operator represents sequential relation. These rules can be implemented by 2 types of state machine: 2-state state machine for rules with no THEN operator and 3-state state machine for rules with 1 THEN operator. Figure 4.6 and Figure 4.7 show examples of these 2 types of state machine.

4.2.2 Experimental Results and Evaluation

All of the data used here was collected from the LISA-P testbed. For this experiment, we used 2 color cameras for upper body pose tracking and 1 color camera for head pose tracking. Real driving data of different drivers was captured and then analyzed by the proposed system. Figure 4.8 shows some results for visual evaluation of the HyHOPE head pose tracking with large range of head rotation, different lighting conditions, and some occlusions. Figure 4.9 shows results of XMOB upper body tracking in several different driving activities. With regard to the run-time performance, we did the analysis on a Pentium(R) D CPU 2.8 GHz and HyHOPE head pose tracking and XMOB upper body tracking ran at around 15 fps (frame per second). Figure 4.10 shows the results of extracting basic activities for each upper body part based on the above pose tracking results. These extraction results are also compared with the manually annotated

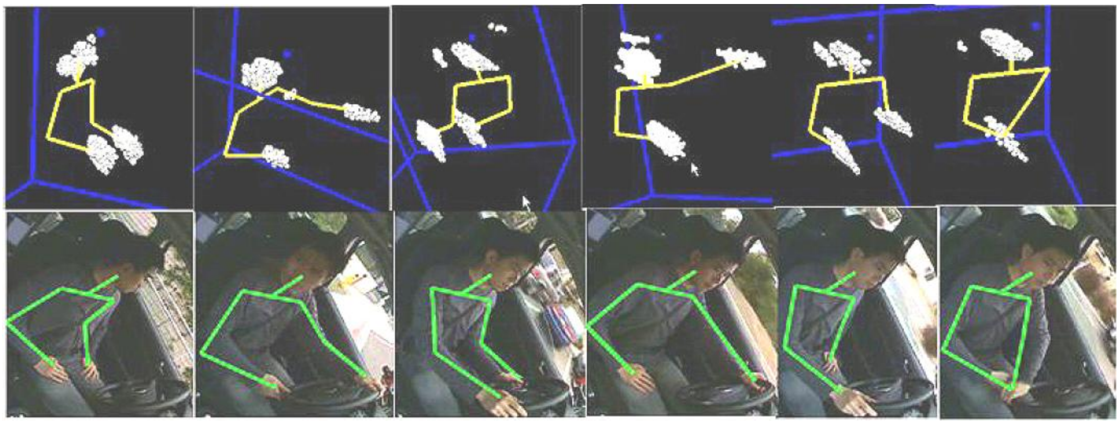


Figure 4.9: Visual evaluation of XMOB upper body tracking in several different driving activities. Top: Upper body pose tracking results in 3-D. Bottom: Superimposed 3-D pose tracking results on image.

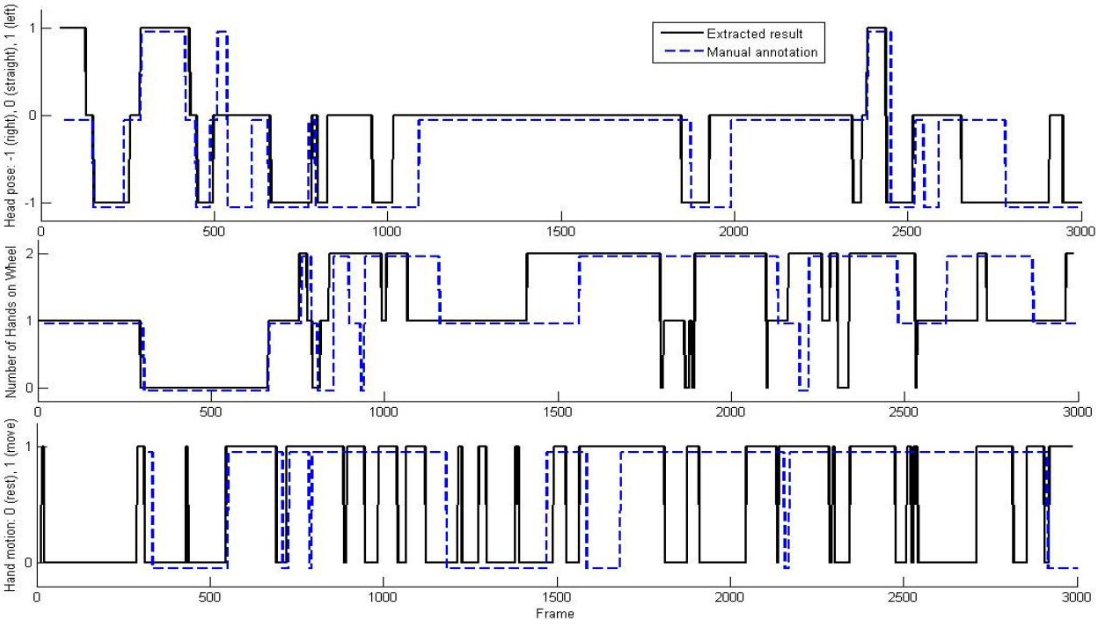


Figure 4.10: Basic activities extraction results. Top: Head (right, straight, or left). Middle: Number of hands on wheel, Bottom: Hand motion (rest or move).

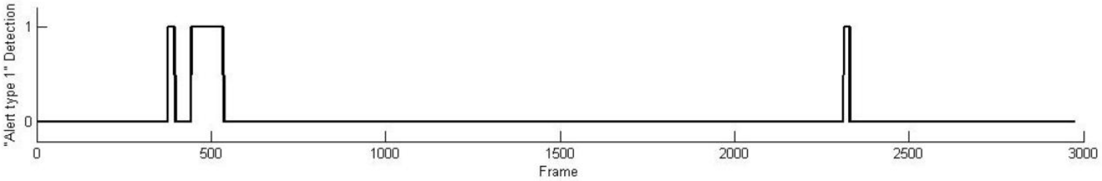


Figure 4.11: Combining basic activities of each upper body part for higher level of semantic description. Result of alert type 1 detection (rule 4 in Figure 4.4D: Head look straight AND No hand on wheel AND Hand rest).

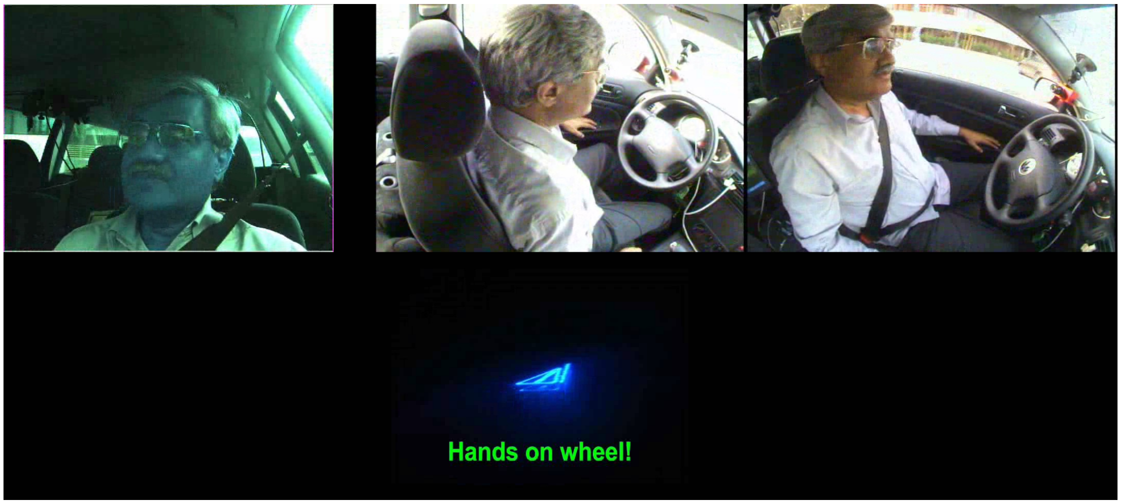


Figure 4.12: Illustration of a system with visual feedback to help the driver in "keeping hands on the wheel and eyes on the road."

ground truth and we see that the proposed system can capture these basic activities quite well. Sample results of combining basic activities of each upper body part for higher level of semantic description are shown in Figure 4.11. Figure 4.12 illustrates a system with visual feedback to help the driver in "Keeping hands on the wheel and eyes on the road."

The evaluation with real-world street driving scene indicates the potential of applying the proposed system to both real-time active safety systems as well as analyzing systems which post process driving data captured from rich contextual realistic situation. For future direction, this driver activity analysis should finally be incorporated with other component of looking at vehicle and surround environment to have a holistic sensing system for intelligent driver support.

4.3 An Optical Flow-based Framework for Driver Foot and Head Behavior Analysis

In this section, we focus on driver foot behavior analysis with applications to intelligent driver assistance. It should be mentioned that an effective driver assistance system needs to be human-centric, and take into account information about all three main components (i.e., environment, vehicle, and driver) interacting in a holistic manner [147, 149]. Among those, driver foot behavior is an important source of information that has a strong impact on vehicle control.

One problem of recent interest to the automotive safety community is that of "pedal misapplication," in which the driver accidentally presses the wrong pedal, as seen in Figure 4.13. Several recent unintended acceleration-related accidents in the U.S. could have been a result of this pedal misapplication phenomenon [48]. Incidents related to pedal misapplication have been observed for many years [110], and the investigation into Toyota's recent "sudden unintended accelerations" [118] has led to a renewed interest in avoiding such incidents. We propose that understanding the driver foot behavior could help to predict and mitigate this kind of problem.

To our knowledge, there are very few research studies in foot gesture and behavior analysis; for example, Choi and Ricci [22] developed a foot-mounted device which can recognize walking gestures. In the domain of driver assistance, some studies related to analyzing driver foot behavior have been published. Park and Sheridan used pressure-based sensors in a driving simulator to show that driver leg motion can help to improve the performance of Antilock Brake System (ABS) [104]. Tanaka et al. [135] analyzed a mechanical model of driver foot and pedal which has potential for having better pedal design and layout. McCall and Trivedi [86] developed a brake assistance system, which took into account both driver's intent to brake and the need to brake given the current situation, in order to determine at what level the driver should be warned. Also in an effort to reduce rear-end collisions, which account for a large portion of traffic accidents [99], Mulder et al. have introduced a haptic gas pedal feedback system for car-

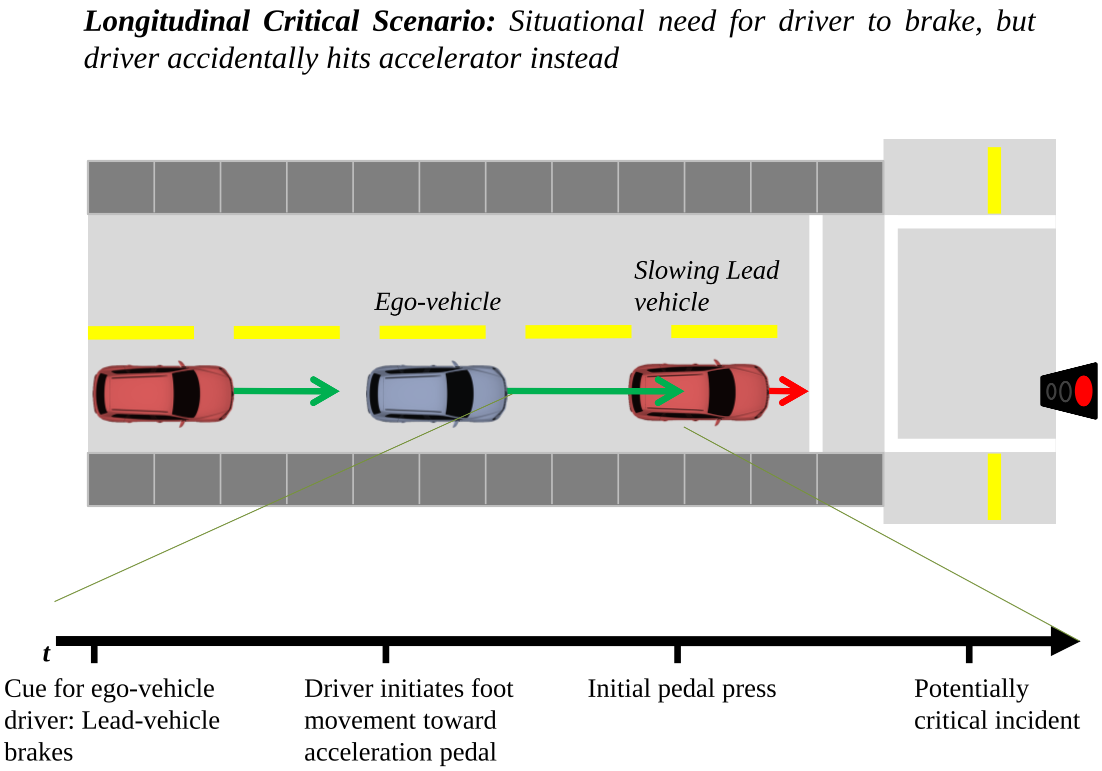

Longitudinal Critical Scenario: Situational need for driver to brake, but driver accidentally hits accelerator instead
Ego-vehicle
Slowing Lead vehicle
t
Cue for ego-vehicle driver: Lead-vehicle brakes
Driver initiates foot movement toward acceleration pedal
Initial pedal press
Potentially critical incident

Figure 4.13: Sample scenario depicting a motivation for foot behavior analysis. In the following research we show that by using vision-based modeling and prediction of foot behavior, we are able to predict instances of "pedal misapplication" at about 200 milliseconds prior to the actual pedal press. This time could provide a critical advantage for an Advanced Driver Assistance System (ADAS) in reducing the severity of a potential collision.

following [94, 2]. In addition to the information of lead-vehicle-separation, they showed that the performance can be improved by using a deceleration control algorithm based on the gas pedal position.

In the following, we develop a new vision-based framework for driver foot behavior analysis. Although embedded vehicle sensor parameters from the Controller Area Network (CAN-bus) like brake or acceleration pedal states tell us something about the foot behaviors, the foot movement before and after a pedal press detected from vision-based sensors can provide valuable information for better semantic understanding of driver behaviors, states, and styles. They can also be used to predict a pedal press before it actually happens. This is very important in time critical (e.g., safety related) situations in which we need time to provide proper assistance to the driver when needed. In the proposed approach, an optical flow-based method is used to track foot movement and a Hidden Markov Model (HMM) is trained to characterize the temporal foot behavior. The vehicle parameters from the CAN-bus are also utilized in live estimation as well as

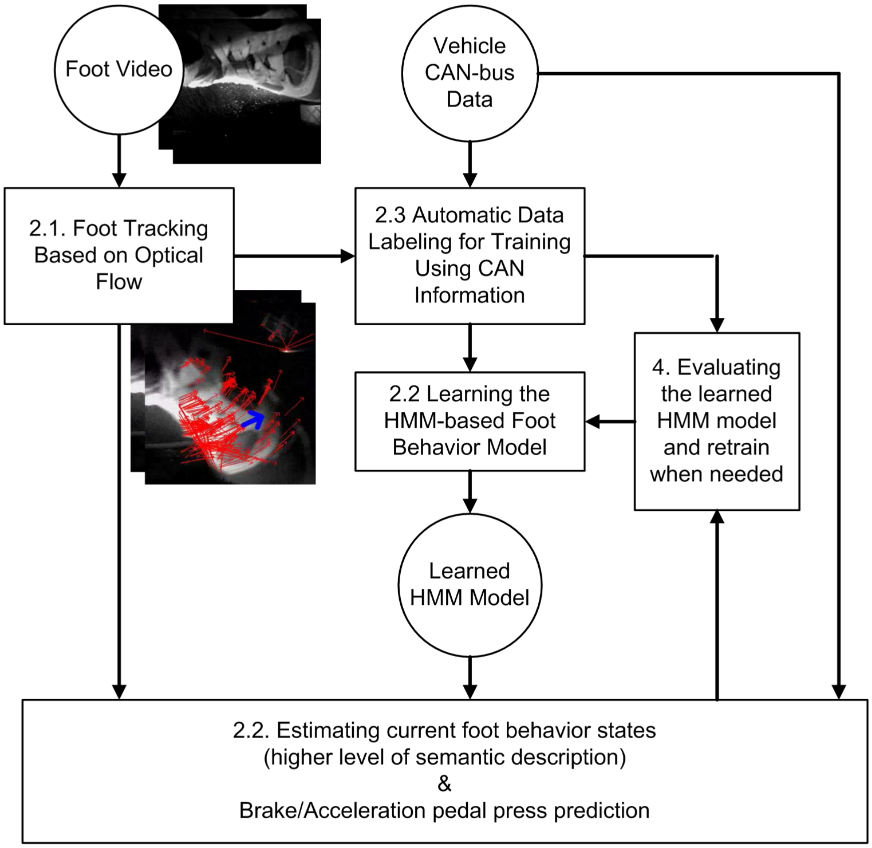

Foot Video
Vehicle CAN-bus Data
2.1. Foot Tracking Based on Optical Flow
2.3 Automatic Data Labeling for Training Using CAN Information
2.2 Learning the HMM-based Foot Behavior Model
4. Evaluating the learned HMM model and retrain when needed
Learned HMM Model
2.2. Estimating current foot behavior states (higher level of semantic description) & Brake/Acceleration pedal press prediction

Figure 4.14: Vision-based framework for driver foot behavior analysis.

in postprocessing, as part of an automatic data labeling procedure for validation purposes. This makes the proposed framework easier to adapt to different subjects and situations and thereby improve performance. The resulting system appears to be the first such system developed for vision-based driver foot behavior modeling and prediction.

4.3.1 Vision-based Framework for Driver Foot Behavior Analysis Using Optical Flow

Our goal is to develop a computer vision system that takes the input of driver foot video, along with vehicle-based pedal sensor measurements, and output a set of higher-level semantic descriptions for the driver foot behavior. It is also desirable if these semantic descriptions can be used to predict a pedal press before it actually happens.

Figure 4.14 shows the components of our proposed vision-based framework for driver foot behavior analysis. First, using data captured from a camera facing the driver's foot, an optical flow-based method is used to track the foot movement. Then we design a Hidden Markov Model (HMM) to learn the temporal foot behavior from the extracted foot movement and vehicle

CAN information. Using the trained HMM, we estimate the current semantic state of the driver foot at each frame as well as use that information to predict a brake of acceleration press before it actually happens. Utilizing reliable information from the vehicle CAN data, we also develop an automatic data labeling procedure so that the learned HMM model can be evaluated in an online manner after each pedal press has occurred. This is an important aspect of the proposed framework since it is easier to adapt to different subjects and situations and therefore potentially improve the performance.

Optical Flow-based Foot Tracking

Optical flow is a well-known computer vision technique for motion estimation. It is based on the assumption of a constant brightness profile

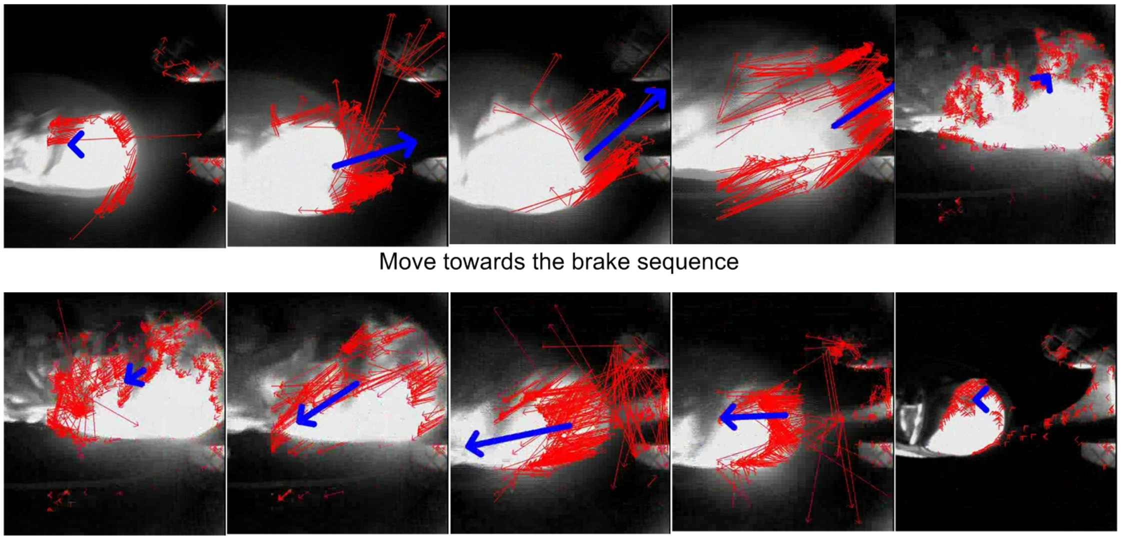
$I(x, y, t) = I(x + u\delta t, y + v\delta t, t + \delta t)$

where (u, v) are the velocities (optical flow) at pixel location (x, y). If we also assume small motion between frames, we can do the Taylor expansion and only keep first-order terms.

$$\begin{aligned} I(x, y, t) + \delta x \frac{\delta I}{\delta x} + \delta y \frac{\delta I}{\delta y} + \delta t \frac{\delta I}{\delta t} &= I(x, y, t) \\ \Rightarrow u\delta t \frac{\delta I}{\delta x} + v\delta t \frac{\delta I}{\delta y} + \delta t \frac{\delta I}{\delta t} &= 0 \\ \Rightarrow u \frac{\delta I}{\delta x} + v \frac{\delta I}{\delta y} + \frac{\delta I}{\delta t} &= 0 \end{aligned}$$

Optical flow has also been applied in several human motion tracking and human gesture analysis studies. For example, Decarlo and Metaxas use optical flow constraint on the motion of a deformable model for face tracking [26]. Holte et al. use a 3-D version of optical flow for view-invariant gesture recognition [50]. In our proposed approach, we use the coarse-to-fine Lucas Kanade algorithm [82] for optical flow detection and combine it with a simple linear motion model for foot tracking. This method works quite well for the driver foot video as visually shown in Figure 4.15. The output of this foot tracking step will provide foot position (p_x, p_y) and velocity (v_x, v_y) at each frame.

Figure 4.16 visualizes all the tracked foot trajectories moving forward from a stopped state to a pedal press, over a single drive (128 stop or go trials). Trajectories towards the brake pedal are marked in red, and trajectories towards acceleration are marked in green. We see that though the end points are quite separate between brake trajectories and acceleration trajectories, the beginnings of those trajectories are more overlapped. Therefore, using information from single points (frames) in this overlapped region to understand driver foot behavior would seem to be ambiguous. In such cases, behavioral models like HMMs that take into account temporal information could help disambiguate these trajectories.



Move towards the brake sequence

Release from the brake sequence

Figure 4.15: Example of optical flow-based foot tracking output for visual evaluation. Red arrows - Detected optical flows. Blue arrow - Tracking output of foot position and velocity at current frame.

HMM-based Foot Behavior Model

By observing the driver foot movement, e.g., Figure 4.15, we see that the the foot motion can be divided into the following semantic states:

1. Neutral (hover off pedal)
2. Moving towards brake pedal
3. Moving towards acceleration pedal
4. Engaging brake pedal
5. Engaging acceleration pedal
6. Release from brake
7. Release from acceleration

Based on this intuitive interpretation, we design a state model for driver foot behavior as shown in Figure 4.17. We see that "clean" pedal press actions would follow the path: Neutral → Move Towards Brake/Accel → Brake/Accel Engaged → Release Brake/Accel → Neutral, and so on. However, there are cases in which after the Release Brake/Accel state, the foot does not actually come back to the Neutral state but has a continuous motion and changes into the next Move Toward Brake or Acceleration state.

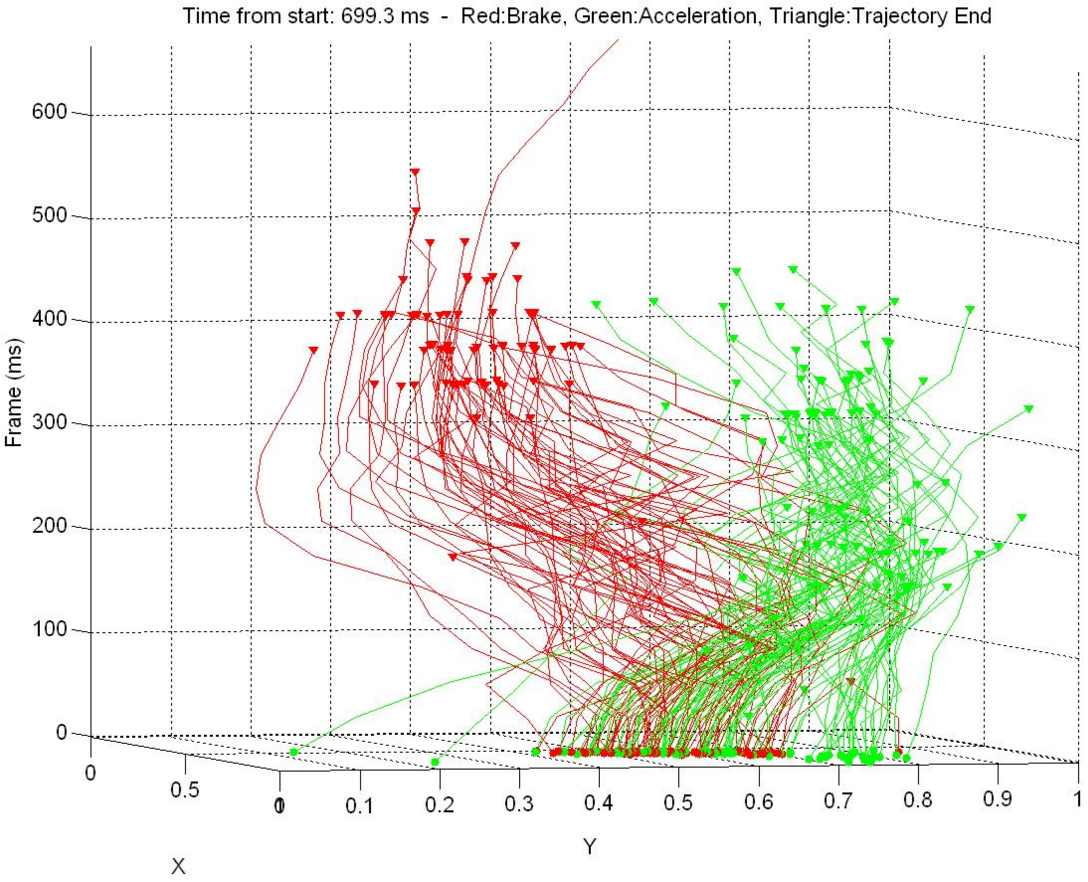


Figure 4.16: Illustration of foot trajectories extracted from optical flow-based foot tracking in a single run; Green: Trajectories towards the acceleration, Red: Trajectories towards the brake. The trajectories are time-aligned to the start of the foot motion, and proceed until a pedal is pressed. The differences between braking and acceleration trajectories can be observed over time.

We choose the HMM-based technique since it can characterize time series data with both spatial and temporal variability. It has been used widely in speech recognition[116] and behavior recognition [108], and recently it has been successfully applied in vision-based gesture recognition, e.g., [74]. To learn the temporal foot behavior, we use a continuous HMM with Gaussian output probability. The elements of our HMM are as follows

- *Hidden states:* We have 7 states $\{s_1, s_2, s_3, s_4, s_5, s_6, s_7\}$ including *Neutral, BrkEngage, AccEngage, TowardsBrk, TowardsAcc, ReleaseBrk, ReleaseAcc.* The state at time t is denoted by the random variable q_t.

- *Observation:* The observation at time t is denoted by the random variable O_t which has 6 components $O_t = \{p_x, p_y, v_x, v_y, B, A\}$ where $\{p_x, p_y, v_x, v_y\}$ are the current position and

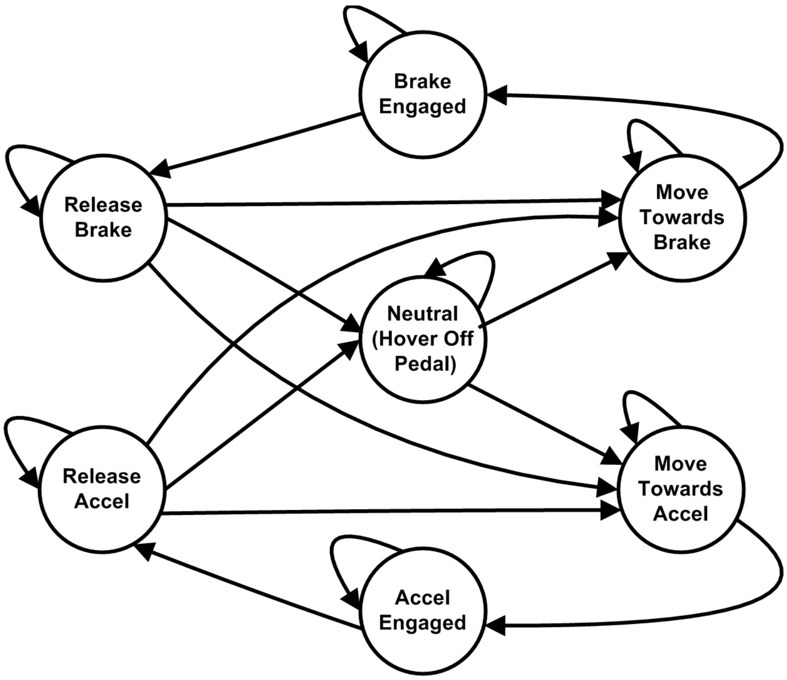


Figure 4.17: Foot behavior HMM state model with 7 states. The states are defined as described in Section 4.3.1.

(optical flow) velocity of driver foot estimated from optical flow-based foot tracking step. $\{B, A\}$ are obtained from vehicle CAN information which determine whether the brake and accelerator are currently engaged or not.

- *Observation probability distributions:* In our HMM model, we assume a Gaussian output probability distribution $P(O_t|q_t = s_i) = N(\mu_i, \sigma_i)$
- *Transition matrix:* $A = \{a_{ij}\}$ is an $7x7$ state transition matrix where a_{ij} is the probability of making a transition from state s_i to s_j

  $a_{ij} = P(q_{t+1} = s_j|q_t = s_i)$
- *Initial state distribution:* We assume an uniform distribution of the initial states.

HMM parameters learning

Given the complete training data (has both observation and hidden states) obtained from the automatic labeling procedure in Section 4.3.1, the set of HMM model parameters Λ including the Gaussian observation probability distribution and the transition matrix can be learned using Baum-Welch algorithm. In our implementation, we use the Probabilistic Modeling ToolKit (PMTK) for Matlab [2] which supports several probabilistic models including HMM.

[2] http://code.google.com/p/pmtk3/

Foot behavior state estimation

Given learned HMM model, at a time t we use the observations in a time window $O_{t-TimeWindow}...O_t$ to estimate the most likely current state for q_t.

$$\hat{s_t} = argmax_{i=1:7}\{P(q_t = s_i|O_{t-TimeWindow}...O_t, \Lambda\} \tag{4.7}$$

Using the current framework, however, in certain cases there is some confusion between $TowardsBrake$ and $BrakeEngage$ as well as between $TowardsAccel$ and $AccelEngage$. In our currently defined $Brk/AccEngage$ states, the foot movement can comprise of different values like no motion, moving forward, and moving backward, which could create some ambiguities in learning the HMM model. Moreover, the current foot camera setup does not cover the whole pedal area. Therefore, when the pedals are pressed a bit hard, part of the foot will be out of the field of view and the optical flow-based foot tracking will not have enough information to provide good tracking output.

However since we have the CAN information to verify if the brake or acceleration is actually engaged or not, this kind of confusion can be corrected with the following practical rule:
If $\hat{s_t} == BrkEngage\ AND\ B_t == 0$ *then* $\hat{s_t} = TowardsBrk$
If $\hat{s_t} == AccEngage\ AND\ A_t == 0$ *then* $\hat{s_t} = TowardsAcc$

Prediction of Brake and Acceleration pedal press

The meaning of our estimated foot behavior states directly connect to the prediction of actual pedal presses. Whenever the foot is in the state *Move Towards Brake* or *Move Towards Acceleration*, we can predict that the corresponding brake pedal or acceleration pedal will be pressed in near future. To avoid possible error noise in the HMM estimation of current foot behavior state, we accumulate over a small time period (a few hundred milliseconds in our experiment) before the current time t.:

$$pb = \sum(\hat{s} = MoveTowardsBrake) \tag{4.8}$$

$$pa = \sum(\hat{s} = MoveTowardsAcceleration) \tag{4.9}$$

Using these accumulation values, the prediction is done as belows

```
IF (pa==0 AND pb==0)
    Prediction = No pedal will be pressed
ELSE
    IF (pa>pb)
        Prediction = Acceleration pedal will be pressed
    ELSE
        Prediction = Brake pedal will be pressed
    END
END
```

Note that although the absolute value and relative proportion between pb and pa have not been exploited in our current implementation, they could give us some useful information about the prediction confidence.

Automatic Data Labeling Based on Vehicle CAN-bus Information

Based on embedded pedal sensor data via the vehicle CAN-bus, we can reliably determine the state (engaged/not engaged) of the brake and acceleration pedals. Utilizing this information combined with the optical flow-based driver foot tracking p_x, p_y, v_x, v_y, B, A, we develop a postprocessing procedure to automatically label the observations into 7 behavior states:

- ***BrakeEngage state:*** Observations with $B = 1$ are labeled as $BrakeEngage$.
- ***AccelEngage state:*** Observations with $A = 1$ are labeled as $AccelEngage$.
- ***Neutral state:*** Observations with $A = 0$ AND $B = 0$ AND $\sqrt{v_x^2 + v_y^2} < MotionThres$ (no motion and off pedal) are labeled as $Neutral$.
- ***TowardsBrake state:*** We detect the events when B changes from 0 to 1 (the start of a BrakeEngage). We look backward in time from these events t_1 until the following happen:
  - A $Neutral$ state is detected at time t_2: Label all the observations between t_2 and t_1 as $TowardsBrake$.
  - A $BrakeEngage$ state is detected at time t_2: This means the foot changed from a previous $ReleaseBrake$ state to $TowardsBrake$ without going through $Neutral$ state (no stop motion), and then to $BrakeEngage$. Therefore, we need to separate between $ReleaseBrake$ and $TowardsBrake$ states. We see that although there is no stop point, the foot trajectory in $ReleaseBrk/Acc$ state and $TowardsBrk/Acc$ state needs to follow opposite directions in x-axis (backward and forward). Therefore, we determine the time t_3 between t_2 and t_1 where p_x is minimum and then label observations between t_2 and t_3 as $ReleaseBrake$ and between t_3 and t_1 as $TowardsBrake$.
  - An $AccelEngage$ state is detected at time t_2: Similarly, we find the time t_3 between t_2 and t_1 where p_x is minimum and then label observations between t_2 and t_3 as $ReleaseAccel$ and between t_3 and t_1 as $TowardsBrake$.
- ***TowardsAccel state:*** We detect the events when A changes from 0 to 1 (the start of an AccelEngage). We look backward in time from these events t_1 until the following happen:
  - A $Neutral$ state is detected at time t_2: Label all the observations between t_2 and t_1 as $TowardsAccel$.
  - A $BrakeEngage$ state is detected at time t_2: Determine the time t_3 between t_2 and t_1 where p_x is minimum and then label observations between t_2 and t_3 as $ReleaseBrake$ and between t_3 and t_1 as $TowardsAccel$.

- An $AccelEngage$ state is detected at time t_2: Determine the time t_3 between t_2 and t_1 where p_x is minimum and then label observations between t_2 to t_3 as $ReleaseAccel$ and between t_3 to t_1 as $TowardsAccel$.

- ***ReleaseBrake state:*** We detect the events when B change from 1 to 0 (the end of a BrakeEngage). We look forward in time from these events t_1 until the following happen:
  - A $Neutral$ state is detected at time t_2: Label all the observations between t_1 and t_2 as $ReleaseBrake$.
  - A $BrakeEngage$ state is detected at time t_2: Determine the time t_3 between t_1 and t_2 where p_x is minimum and then label observations between t_1 to t_3 as $ReleaseBrake$ and between t_3 to t_2 as $TowardsBrake$.
  - An $AccelEngage$ state is detected at time t_2: Determine the time t_3 between t_1 and t_2 where p_x is minimum and then label observations between t_1 and t_3 as $ReleaseBrake$ and between t_3 and t_1 as $TowardsAccel$.

- ***ReleaseAccel state:*** We detect the events when A change from 1 to 0 (the end of an AccelEngage). We look forward in time from these events t_1 until
  - A $Neutral$ state is detected at time t_2: Label all the observations between t_1 and t_2 as $ReleaseAccel$
  - A $BrakeEngage$ state is detected at time t_2: Determine the time t_3 between t_1 and t_2 where p_x is minimum and then label observations between t_1 and t_3 as $ReleaseAccel$ and between t_3 and t_2 as $TowardsBrake$.
  - An $AccelEngage$ state is detected at time t_2: Determine the time t_3 between t_1 and t_2 where p_x is minimum and then label observations between t_1 and t_3 as $ReleaseAccel$ and between t_3 and t_1 as $TowardsAccel$.

The output of this automatic procedure was qualitatively validated by looking at the labeled states over time in a synchronized mode with video of the foot, and the labeled data looked reasonable. Note that this automatic labeling procedure need to use a whole long data sequence where we can look into the "past" and "future" of a pedal press event. In the estimation and prediction mode, we only have a $Time-Window$ of previous (past) observations. Therefore, we need to learn the HMM model for that purpose.

4.3.2 Experimental Results and Evaluation

Since the proposed postprocessing framework can automatically label the data, the HMM parameters can be learned specifically for different subjects for better performance. Therefore, in this section, we will analyze the behavior state estimation and pedal press prediction for 2 different subjects (subjects 1 and 2, using data from all three conditions of the experiment) to show performance of the proposed framework. We also test a subject-wise cross-validation

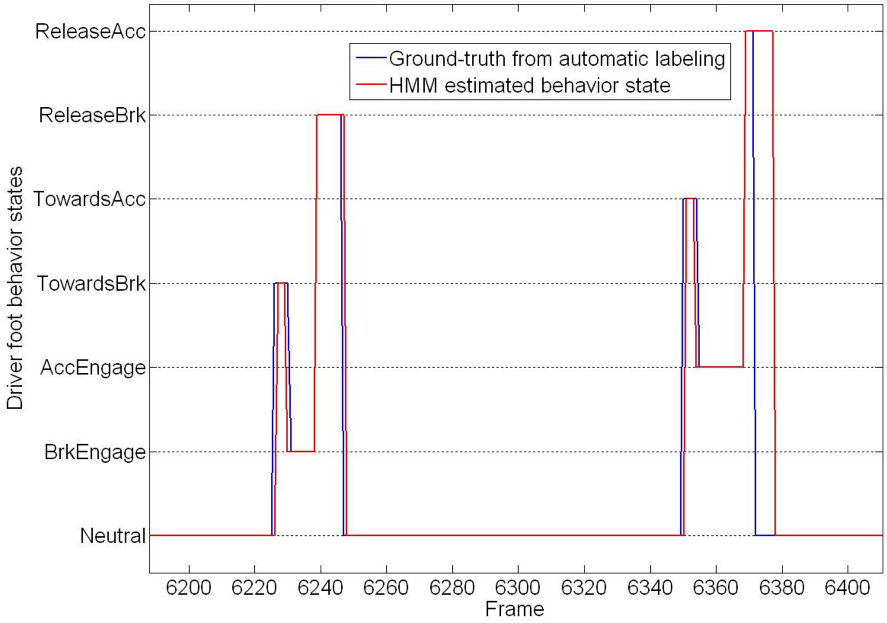


Figure 4.18: Plot of HMM estimated behavior states compared to the ground truth obtained from automatic labeling.

Table 4.3: Confusion matrix of 7 behavior states averaged over 15 runs (each subject does 3 runs with visual cues only, audio cues only, and both audio and visual cues). For each run, the first 2 minutes are used for training and ∼ 8 remaining minutes are used for testing. Each row sums up to 100%.

	Predicted State						
Actual	Neutral	BrkEng	AccEng	TwdBrk	TwdAcc	RlsBrk	RlsAcc
Neutral	**0.9034**	*0*	*0*	*0.0087*	*0.0254*	*0.0199*	*0.0426*
BrkEng	*0*	**0.9952**	*0*	*0.0045*	*0*	*0*	*0.0003*
AccEng	*0*	*0.0009*	**0.9955**	*0*	*0.0027*	*0.0003*	*0.0007*
TwdBrk	*0.0912*	*0*	*0*	**0.8762**	*0.0175*	*0.0009*	*0.0143*
TwdAcc	*0.1181*	*0.0006*	*0*	*0.0090*	**0.8664**	*0*	*0.0059*
RlsBrk	*0.0371*	*0*	*0.0007*	*0.0007*	*0.0006*	**0.9596**	*0.0014*
RlsAcc	*0.0082*	*0.0037*	*0*	*0.0098*	*0.0049*	*0.0057*	**0.9678**

Table 4.4: The recall and precision of pedal press prediction (averaged over 15 runs, ∼ 10 minute each)

Time before actual pedal press		**Brake prediction**		**Acceleration prediction**	
Frame	Milliseconds	Precision	Recall	Precision	Recall
30	999	1.0000	0.0458	0.8980	0.1042
15	499	0.8016	0.0532	0.9153	0.1412
10	333	0.7947	0.0854	0.8882	0.3246
8	266	0.8869	0.2284	0.9065	0.5264
7	233	0.9010	0.3085	0.9221	0.6265
6	200	0.9037	0.3972	0.9352	0.7265
5	167	0.9220	0.5090	0.9398	0.7841
4	133	0.9564	0.6561	0.9529	0.8266
3	100	0.9695	0.8111	0.9630	0.8622
2	67	0.9781	0.8816	0.9694	0.8971
1	33	0.9823	0.8970	0.9732	0.9138
0	0	1	1	1	1

procedure (i.e., train on subject 1 and test on subject 2) to illustrate the ability of the proposed framework to adapt the learned model to different subjects and situations.

HMM Foot Behavior State Estimation

For each experimental run of a subject (about 10 minutes of data), we use the first 2 minutes for training and the remaining part for testing. Figure 4.18 visualizes an example of the estimated foot behavior state in comparison with the ground truth from automatic data labeling procedure.

The confusion matrix of 7 states which is averaged over 15 runs (each subject does 3 runs with visual cues only, audio cues only, and both audio and visual cues) is shown in Table 4.3. The mean of correct classification rate for the 7 classes is 93.77%. The significant source of confusion remains mostly in the distinction between the end of a "Neutral" state and transition to a movement "Towards" the pedals. The identification of the initial movement point is more difficult in live analysis than in postprocessing. Examples of this can be seen in Figure 4.18, which shows a consistent pattern of slightly later transitions out of "Neutral" in the predicted behavior states.

Brake/Acceleration Pedal Press Prediction

In applying this state detection methodology to pedal press prediction, we would like to accurately predict the pedal presses as soon as possible. However, typically, there is a trade-off between the time advantage of prediction (how soon is the time of prediction compared to the

Table 4.5: Confusion matrix of 7 states with leave one out cross validation. Each row sums up to 100%. For each subject, we train the HMM behavior model with data from the other subjects and test on the selected one. The results are then averaged over all tests.

	Predicted State						
Actual	Neutral	BrkEng	AccEng	TwdBrk	TwdAcc	RlsBrk	RlsAcc
Neutral	**0.9066**	*0*	*0*	*0.0114*	*0.0359*	*0.0200*	*0.0262*
BrkEng	*0*	**0.9022**	*0*	*0.0978*	*0*	*0*	*0*
AccEng	*0*	*0*	**1**	*0*	*0*	*0*	*0*
TwdBrk	*0.1697*	*0*	*0*	**0.6701**	*0.1532*	*0*	*0.0069*
TwdAcc	*0.1951*	*0*	*0*	*0.0110*	**0.7940**	*0*	*0*
RlsBrk	*0.1432*	*0.0002*	*0.0002*	*0*	*0.0590*	**0.6267**	*0.1709*
RlsAcc	*0.1086*	*0.0022*	*0*	*0.0113*	*0*	*0.0066*	**0.8713**

actual pedal press) and both the recall [3] and precision rates[4]. To analyze this kind of trade-off, we attempt to make predictions at various points leading up to an actual pedal press (as determined from embedded pedal sensor data). For example, we may set a threshold time of 300ms before the pedal press, and using the accumulated information leading up to that time (equation (4.8)), we can determine the performance of a corresponding predictive classifier (i.e., classifying whether/which pedal will be pressed).

Tables 4.4 shows the precision and recall of brake and acceleration predictions with different thresholds of time before the actual press. These results are the average over all 15 runs. We see that at 133 ms prior to the actual pedal press, a major part ~74% of the pedal presses can be predicted (there is 82.66% recall rate for acceleration predictions, and 65.61% for brake predictions).

Validation Across Subjects

We use leave one out cross validation to analyze the performance of the learned HMM behavior model across subjects. For each test, we select a subject to test on and train the HMM behavior model with data from the remaining subjects. The results are then averaged over all tests as shown in Table 4.5. We see that the performance degrades in comparison to Table 4.3 where the HMM model is trained and tested on the same subject for each test. There are more confusions especially in the *TowardsBrake*, *TowardsAccel*, and *ReleaseBrk* states, which further indicates that there are some differences in foot movement style between subjects. This demonstrates the importance of the automatic data labeling procedure which makes it easier to train individual models for each of the subjects in order to achieve better performance.

[3] recall = $\frac{|\{Actual\_pedal\_presses\} \cap \{Predicted\_pedal\_presses\}|}{|\{Actual\_pedal\_presses\}|}$

[4] precision = $\frac{|\{Actual\_pedal\_presses\} \cap \{Predicted\_pedal\_presses\}|}{|\{Predicted\_pedal\_presses\}|}$

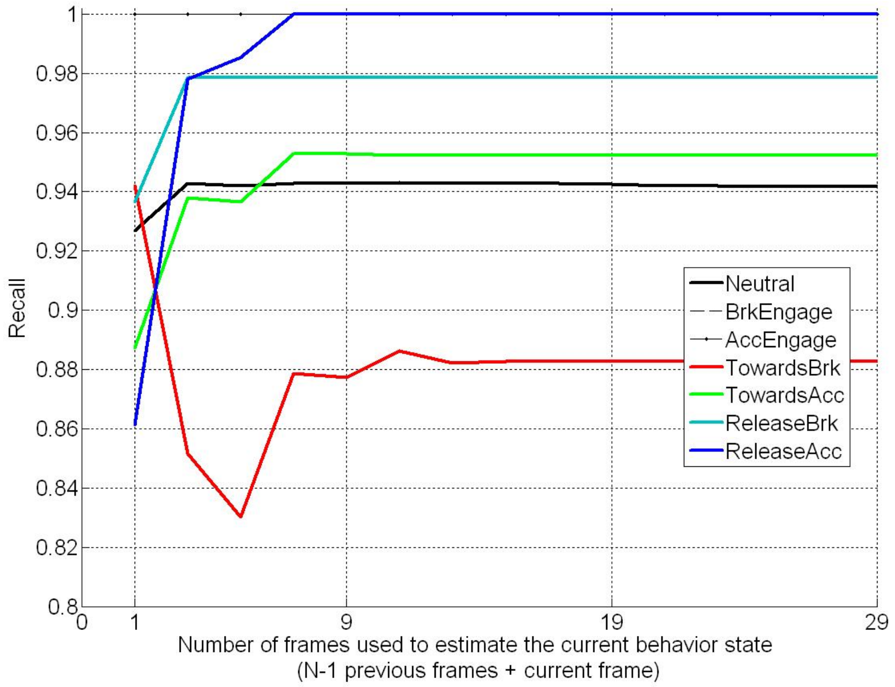


Figure 4.19: Effect of changing observation time window for estimating the foot behavior states (illustrated for Subject 1).

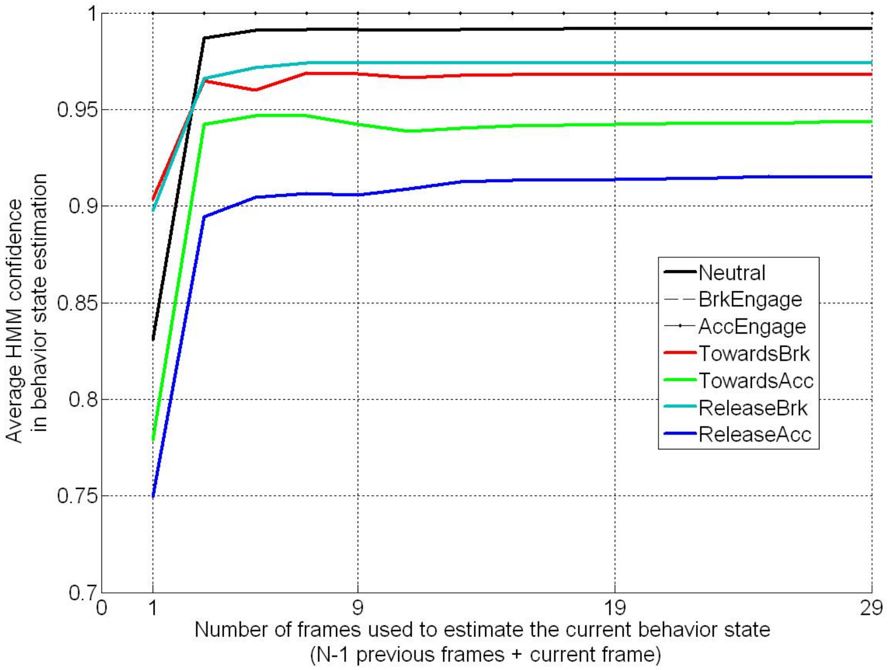


Figure 4.20: HMM confidence in estimating behavior states with different observation time windows (measured by $argmax_{i=1:k}\left\{P(q_t = S_i|O = O_{t-TimeWindow}...O_t, \Lambda\right\})$.

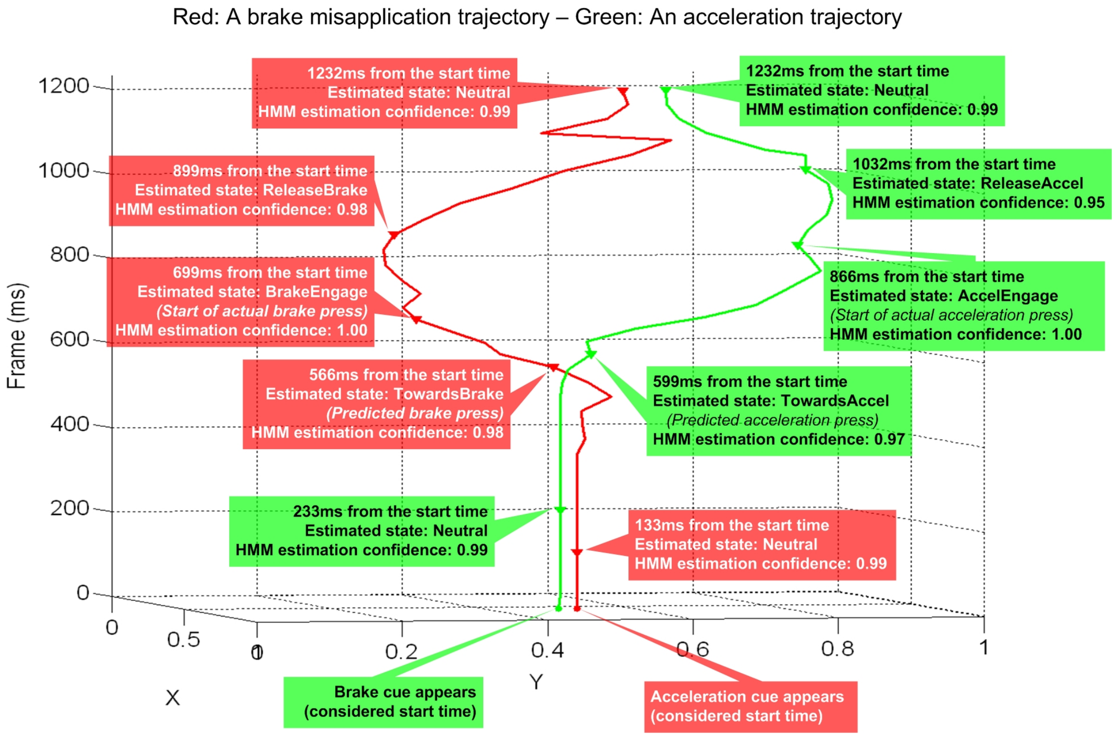


Figure 4.21: Trajectories of an actual brake misapplication (in red - the subject was cued to hit acceleration but instead applied brake) and an acceleration misapplication (in blue - the subject was cued to hit brake but instead applied acceleration). The trajectories are obtained by optical flow-based foot tracking (the X, Y axes are the image coordinates of the tracked foot). The labeled points show the outputs of the HMM-based foot behavior analysis. In each case, the pedal misapplication is correctly predicted over 100ms in advance of the actual pedal press.

Benefit of Temporal Information

Another analysis of interest is the relative contribution of temporal information, in comparison with the instantaneous observation information at each single frame. Figure 4.19 shows the recall rate for 7 states (the diagonal of the confusion matrix) as a function of the observation time window used for estimation (a time window of 1 means only the observation at current frame is used). We see that for most of the states the temporal information did help to improve the performance. Figure 4.20 also shows that with a few historical frames, the average confidences of the learned HMM model estimating the foot behavior state are all over 90% (the confidence is determined by the max state estimation probability in equation (4.7)).

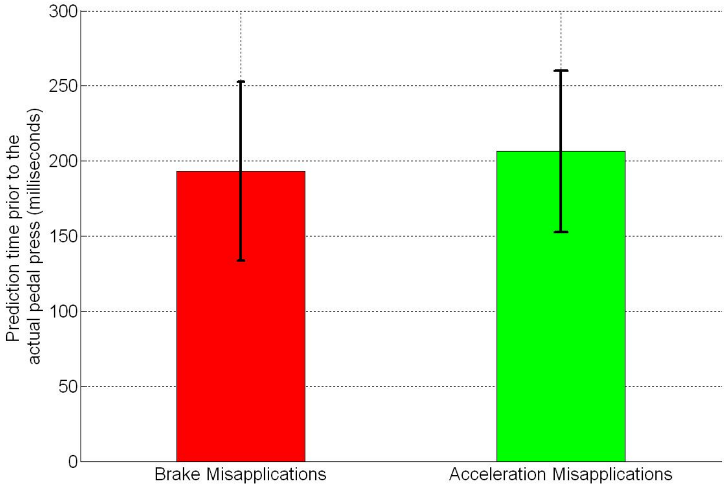


Figure 4.22: Statistics (mean and standard deviation) of the prediction time prior to the actual pedal press for brake misapplications (the subject was cued to hit acceleration but instead applied brake) and acceleration misapplication (the subject was cued to hit brake but instead applied acceleration). Over 15 runs ($\sim$ 10 minute each), there were 5 brake misapplications and 15 acceleration misapplications.

Application of Foot Gesture Analysis: Prediction of Pedal Misapplication

As mentioned above, one potential application for the detection of foot behavior is in predicting and mitigating the effects of pedal misapplication. Due to the nature of the experimental data, we were able to observe several instances of unintended pedal presses, or "misapplications." In these cases, the subjects were cued to hit a specific pedal but instead applied the wrong pedal. These cases tended to occur when the subjects had a significant workload to deal with (e.g., environmental stimuli), or a rapid set of alternating cues to respond to (e.g., confusing historical context [30]).

Figure 4.21 visualizes the outputs of the proposed framework for a brake misapplication example (when the subject was cued to hit acceleration but instead applied brake) and an acceleration misapplication example (when the subject was cued to hit brake but in instead applied acceleration). Over 15 runs, there were 5 cases of brake misapplications and 15 cases of acceleration misapplications. In all these twenty cases, the movement towards the wrong pedal was correctly predicted by the HMM prediction framework. The average prediction time prior to the actual pedal press was $\sim$193 milliseconds for brake misapplications and $\sim$206 milliseconds for acceleration misapplications. The statistics of these prediction times are shown in Figure 4.22.

In real-world instances, if an intelligent ADAS is aware that a driver needs to begin applying the brakes, but instead detects a move towards the accelerator, the ADAS could take measures to reduce the effects of the error. Haptic feedback, either alerting the driver or making

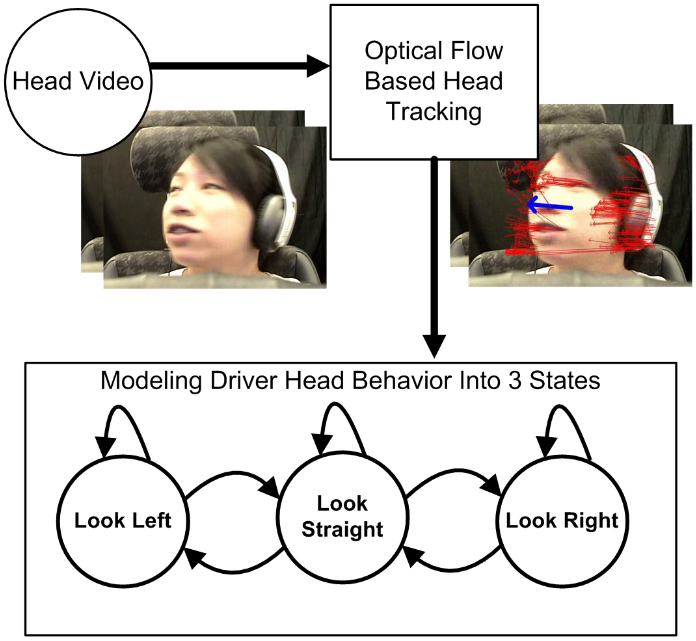

Head Video
Optical Flow Based Head Tracking
Modeling Driver Head Behavior Into 3 States
Look Left
Look Straight
Look Right

Figure 4.23: Framework for tracking driver head with optical flows and modeling head behavior into 3 states.

it more difficult to press the pedal, could possibly prevent a simple error from escalating into a critical incident.

4.3.3 Extending The Optical Flow-based Framework for Driver Head Behavior Analysis

In many driver assistance systems such as [29, 85, 139], determining whether the driver is looking straight, looking left, or looking right could already provide useful information about the driver visual attention. We see that a similar framework for driver foot could be applied for driver head behavior analysis as shown in Figure 4.23. Although we have used the commercial FaceLAB head/eye tracker in our experiments, we would like to compare the performance of using FaceLAB output vs. using optical flow head tracking output in classifying driver head behavior into 3 states $\{LookStraight, LookLeft, LookRight\}$.

Figure 4.24 shows an example of detecting the start and end of a head turn based on the head rotation output from FaceLAB vs. based on optical flow head movement. To compare the performance, we run the head turn detection using FaceLAB and optical flow tracking for 6 subjects in the experiment described in Section 5.2.2. In the 3 runs AD, VD, AVD of these subjects, there are 833 head turns in total. Among those, the agreement between head turn detection using FaceLAB and using optical flow tracking is over 92%. Using our visualization

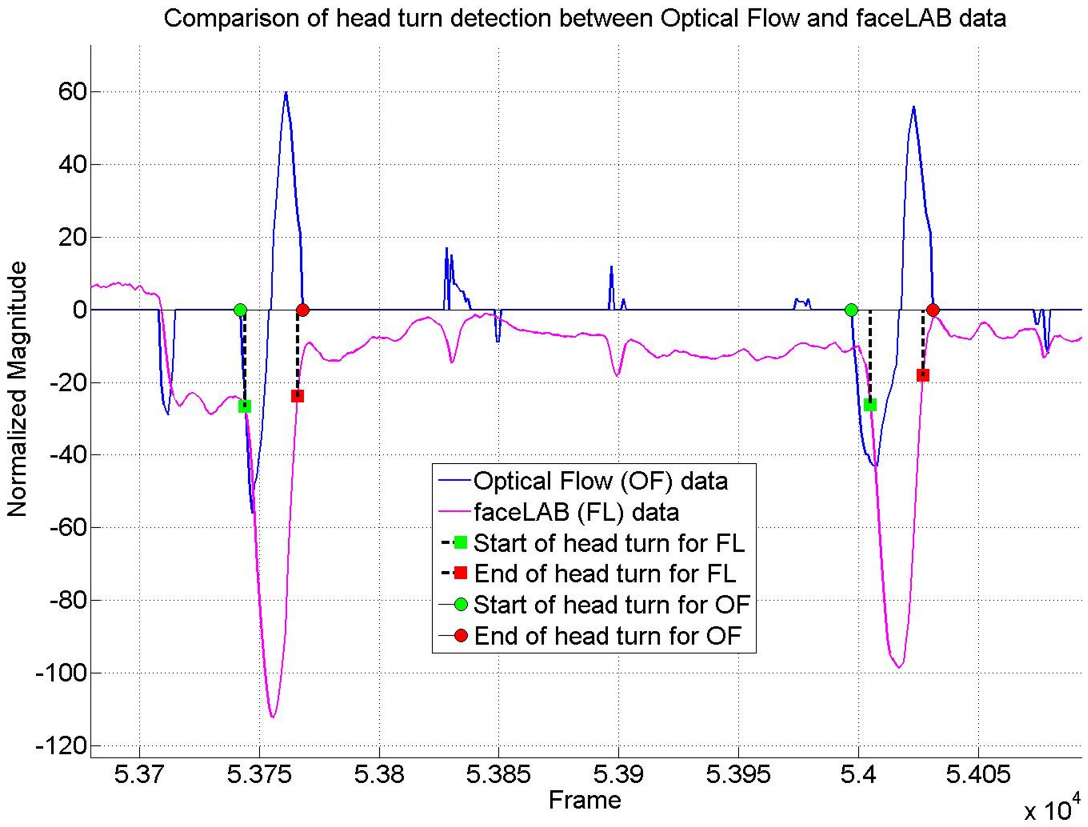


Figure 4.24: Detection of head turns (start and end points in time) using optical flow head tracking output vs. output of the commercial FaceLAB system.

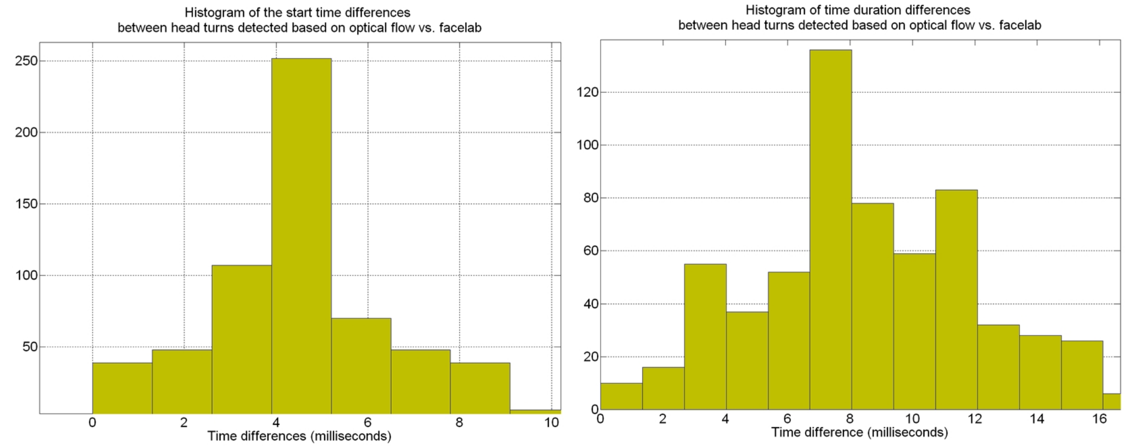


Figure 4.25: Comparison between head behavior detected based on optical flow head tracking vs. the commercial FaceLAB system.

toolbox, we see that the main reason for this difference is due to the cases where there are hand movements (on the steering wheel) in the scene which are misdetected as head movements. Among 92% agreement of head turn detections, we compare the start time and the duration (time from the start to end point of a head turn) of detected head turns. As shown in Figure 4.25, the differences are small (in term of 10ms). Therefore in classifying driver head behavior into 3 states $\{LookStraight, LookLeft, LookRight\}$, we could use optical flow tracking which requires much lower cost compared to the commercial FaceLAB system.

4.3.4 Discussion

We have proposed and implemented a new vision-based framework for driver foot and head behavior modeling and prediction, which is an important but still open area in developing intelligent driver assistance systems. To our knowledge, the proposed system is the first system developed for vision-based semantic understanding of driver foot behavior, including both modeling and prediction. Using the output of optical flow-based foot tracking combined with embedded pedal sensor information, we design an HMM model to learn the temporal foot behavior. The learned HMM model is then used to interpret driver foot movement into 7 behavior states (*Neutral, BrakeEngage, AccelerationEngage, TowardsBrake, TowardsAcceleration, ReleaseBrake, and ReleaseAcceleration*) as well as to predict a brake or acceleration press before it actually happens. The proposed framework also utilizes reliable vehicle sensor information for an automatic post-hoc data labeling procedure. An opportunity in follow-up research is to use this labeled data to update the HMM model for different subjects and situations in an online manner.

In our experiment with real-world driving data, the proposed framework provided good results with a recall rate of ∼94% average over all 7 behavior states in estimating foot behavior states compared to the labeled ground truth. The analysis of different observation time windows showed that using temporal information did help to improve the performance. Results of our cross subject test (i.e., training the HMM with one subject and test it on the other subject) implied that different drivers may have different foot behavior styles. Therefore, the potential of online updating the HMM behavior model for different subjects and situations is an important aspect of the proposed framework. With regard to the pedal press prediction based on the estimated behavior states, a major portion of the pedal presses can be precisely predicted before they actually happen (e.g., recall rate of ∼74% at 133ms before the actual press). This indicates the potential of using the proposed framework in some open problems in intelligent driver assistance, like predicting and mitigating the pedal misapplication phenomenon.

It should be mentioned that ∼133ms seems not to be enough to provide an useful visual feedback, but it might be possible to generate an appropriate haptic pedal feedback. As reported in [2], the response time delay for visual feedback is around 200 - 500ms, while responses to continuous haptic feedbacks are significantly faster - on the order of ∼50ms, with less cognition involved. Given that we are able to detect several instances of pedal misapplication about 200 milliseconds prior to the pedal press, we have demonstrated that the approach is a feasible tactic to help avoid dangerous situations.

When considering real-world driver assistance applications, the system processing time is also an important factor that needs to be taken into account. In our implementation, the time of HMM behavior state estimation is not an issue since it typically takes only few milliseconds for each estimation. The current bottleneck is the computation of the optical flow-based foot tracking. It runs at about 10 frames per second, implying a delay time for optical flow computation of up to 100ms - which is slow for time critical applications. There is potential to make this approach more feasible to real-world driver assistance application, for example by using parallel algorithms for optical flow. We can also reduce the image resolution and the number of image features used for optical flow estimation (currently we use 640x480 image resolution and 200 image features). This kind of reduction however might lead to some trade-offs in the quality of the output that we will need to consider.

The implications for improving the safety and comfort of driving are significant. By proposing a novel system that is able to model and predict foot behavior in vehicles, we have ultimately demonstrated a unique opportunity to harness computer vision in improving safety on our road.

4.4 Acknowledgments

Chapter 4 is based on material that is published in the IEEE Transactions on Industrial Informatics (2012), IEEE International Conference on Vehicular Electronics and Safety (2009) both by Cuong Tran and Mohan M. Trivedi, and Computer Vision and Image Understanding — Elsevier (2012) by Cuong Tran, Anup Doshi, and Mohan M. Trivedi. The dissertation author was the primary investigator and author of these papers. This research is sponsored in part by UC Discovery Program and National Science Foundation. We would also like to thank the volunteers and our colleagues in the CVRR lab for their valuable advices and assistance.

Chapter 5

Driver Behavior Study with Multimodal Testbeds and Experiments

One main targeted application domain of this dissertation is developing intelligent driver assistance systems which have an increasingly important role nowadays. World Health Organization reported that traffic collisions account for about 1.2 million fatalities and over 20 million injuries worldwide each year [102]. It is also reported that a large portion of accidents (80% of crashes and 65% of near crashes) is caused by human errors like driver inattention or cognitive overload. The lives lost and significant costs associated with traffic collisions require new approaches to reduce the number and effects of such incidents. While passive safety technologies like airbags and seatbelt can help to limit the injuries during collision, novel preventative technologies (active safety) which can help to predict crashes in advance to avoid them are desirable. To be effective, such Intelligent Driver Assistance Systems (IDAS) need to be human centric and work in a holistic manner which takes into account different components including sensors looking at the environment (e.g., the roads, other cars), looking at the vehicle (e.g., looking at steering angle, vehicle speed), as well as looking at the driver [149].

There have been a large amount of research studies in the related area [76]. However, to develop an efficient IDAS, which can understand and assist the driver in a nonintrusive and naturalistic manner, is still an open question. As illustrated in Figure 5.1, since human normally communicate through different channels such as text, audio, and visual information, an effective IDAS should also perceive and interact with the driver using multimodal signals. Typically, a driver assistance system is a complex system which requires processing of information from variety sources including the ego vehicle (e.g., current speed, acceleration), driving environment

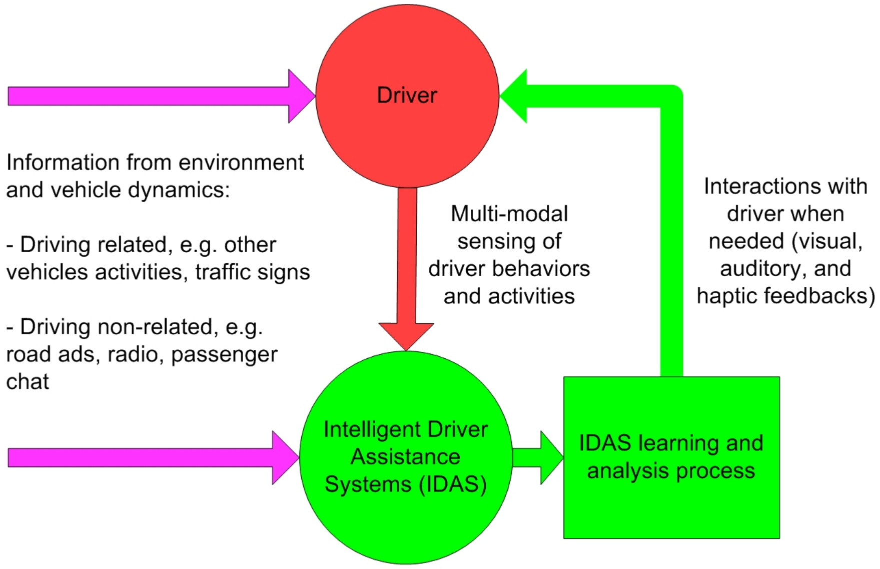

Driver
Information from environment and vehicle dynamics:
- Driving related, e.g. other vehicles activities, traffic signs
- Driving non-related, e.g. road ads, radio, passenger chat
Multi-modal sensing of driver behaviors and activities
Interactions with driver when needed (visual, auditory, and haptic feedbacks)
Intelligent Driver Assistance Systems (IDAS)
IDAS learning and analysis process

Figure 5.1: Flowchart showing the interaction between drivers and Intelligent Driver Assistance Systems (IDAS's).

(e.g., other vehicles, obstacles, traffic signs), and driver (e.g., driver behavior and cognitive state). Therefore, one initial and major difficulty in studying and developing efficient IDAS's is to have adequate infrastructure testbeds which can be used to create and capture rich information about different driving scenarios for analysis while still preserve the safety of the experimental subjects.

In this chapter, we introduce two multimodal driving testbeds (a real-world vehicle and a driving simulation) that we have been building and maintaining in our Computer Vision and Robotics Research (CVRR) lab at UC San Diego as well as several joint audio-visual experiments and databases that we have developed for studying traffic accident prevention. These experiments focus on two typical types of traffic accident scenarios which are longitudinal accidents (i.e., head-on and rear-end collisions) and lateral accidents (i.e., side collisions). Figure 5.2 shows samples of these two critical scenarios. Related to the lateral critical scenario, one of our objectives is to understand and support driver spatial awareness using multimodal signals. Related to the longitudinal critical scenario, we focus on the phenomenon of pedal errors (or "unintended acceleration"). Incidents related to pedal errors have been observed for many years [110], and in many cases involving fatal accidents. Recently, several unintended acceleration-related accidents in the U.S. could have been a result of this pedal misapplication error phenomenon [48] and the investigation into Toyota's recent "sudden unintended accelerations" [118] has led to a renewed interest in understanding and avoiding such incidents. Based on these experiments, we will also discuss our analysis which provide insight into the effect of audio and visual cues on driver behavior such as the reaction time, movement time, and number of pedal errors.

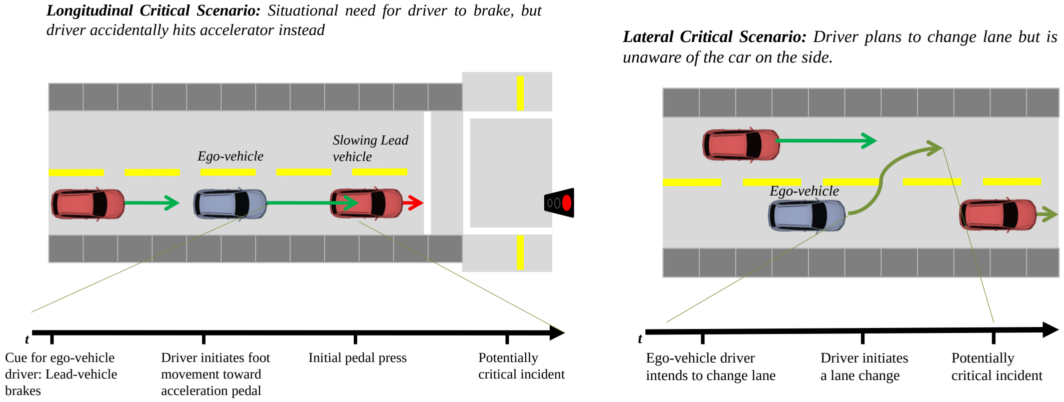

Longitudinal Critical Scenario: Situational need for driver to brake, but driver accidentally hits accelerator instead
Ego-vehicle
Slowing Lead vehicle
t
Cue for ego-vehicle driver: Lead-vehicle brakes
Driver initiates foot movement toward acceleration pedal
Initial pedal press
Potentially critical incident
Lateral Critical Scenario: Driver plans to change lane but is unaware of the car on the side.
Ego-vehicle
t
Ego-vehicle driver intends to change lane
Driver initiates a lane change
Potentially critical incident

Figure 5.2: Sample scenarios depicting the critical situations related to longitudinal accidents (left) and lateral accident (right).

5.1 Multimodal Driving Testbeds

Typically, a coordination between real-world driving testbed and simulation driving testbed is needed. Working with real world driving testbed is important and is the ultimate goal. However, simulation environment can provide more flexibility in configuring sensors and designing experiment tasks for deeper analysis which might be difficult and unsafe for implementing in real world driving. We see that observations from real world driving data can help on initiating the design of simulation experiment. Then this simulation experiment can be modified and improved to achieve several desired analyses. However, there are always gaps between simulation environment and real world which happens even with nowadays complex and expensive simulations (e.g., the realistic feel of vehicle dynamics and surround environment). Therefore, the usefulness of analyses and findings in simulation environment should again be verified with the real world driving data. We take this kind of coordination into account when developing our driving testbeds.

5.1.1 Real-world Driving Testbed LISA-P

Figure 5.3 shows our real-world driving testbed LISA-P, a Volkswagen Passat.

- Sensor Inputs: Information about the vehicle dynamics (e.g., speed, acceleration, pedal positions, steering wheel angle) can be read through the CAN-Bus (Controller Area Network) capture interface. LISA-P was designed with place holders and wire connections for installation of multiple cameras and microphones. We have used two microphones for audio capture and 5 cameras for video capture (2 for driver upper body, 1 for driver face, 1 for driver foot, and 1 stereo camera looking forward to the environment). All sensor inputs are captured into a single computer in a synchronized manner.

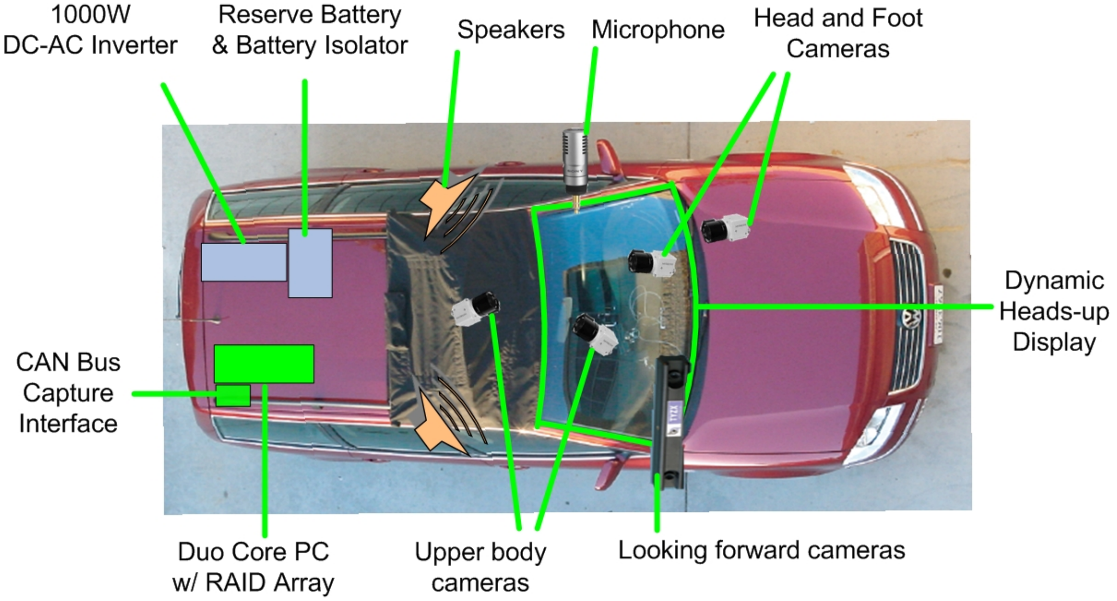


Figure 5.3: LISA-P real-world driving testbed with multimodal sensing and displays.

- Feedback Types: LISA-P was instrumented with a novel laser-based see-through windshield display where we can display visual feedback to the driver [28]. We also have speakers to provide auditory feedback.

5.1.2 Simulation Driving Testbed LISA-S

Our simulation driving testbed LISA-S is shown in Figure 5.4. The driving simulator was developed based on the open source TORCS project.[1] With this simulator, we can design different road tracks and surrounding scenes as well as add other vehicles and have them behave in some predefined manners (which would be very hard to have in the real-world environment).

- Sensor Inputs: In LISA-S, we have a dedicated (commercial) eye/head tracking system FaceLAB. We have cameras for visual inputs (looking at driver head, foot, and upper body), microphones for audio inputs, and a skin conductance sensor for physiological signals. Related information from the driving simulator (e.g., the dynamics of ego vehicle and other vehicles, track information) are also captured. Sensor inputs in LISA-S are captured on different computers (one for the dedicated eye tracker, one for audio-visual and physiological signals, and one for simulation environment). The synchronization between computers is done by having synchronization signals sent frequently to all the computers.

[1] http://torcs.sourceforge.net/index.php

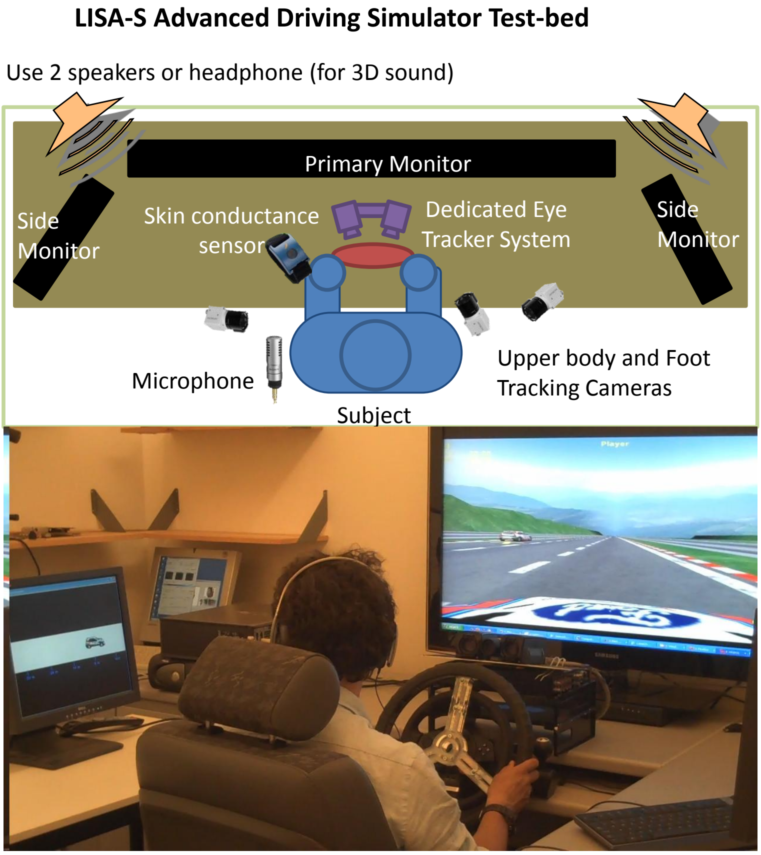

LISA-S Advanced Driving Simulator Test-bed
Use 2 speakers or headphone (for 3D sound)
Primary Monitor
Side Monitor
Skin conductance sensor
Dedicated Eye Tracker System
Side Monitor
Microphone
Upper body and Foot Tracking Cameras
Subject

Figure 5.4: LISA-S simulation driving testbed with multimodal sensing and displays.

- Feedback Types: Beside the main monitor for the driving simulator, LISA-S has two side monitors for showing spatial information or distractions when needed. There are speakers for audio feedback, and a headphone is also used when 3-D audio cues need to be provided to the driver.

5.2 Joint Audio-visual Driving Experiments and Databases

In this section, we will describe 3 joint audio-visual experiments and databases that we have developed based on the LISA-P and LISA-S testbeds. These experiments focus on studying two typical types of traffic accident which are longitudinal and lateral critical scenarios (Figure 5.2).

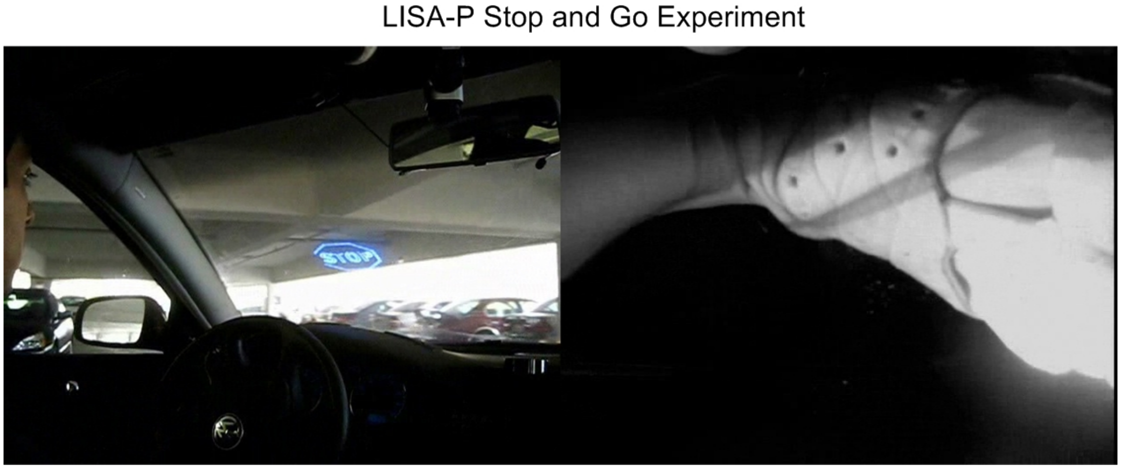


Figure 5.5: LISA-P Stop-and-Go experiment

5.2.1 Real-world Stop-and-Go Driving Experiment in LISA-P

This experiment was designed to approximate a stop-and-go traffic, in order to maximize the number of pedal presses observed per participant, and to study driver behavior in sequential contexts. This is actually one of a series of experiments in cooperation with a cognitive scientist to study "sequential effects" in complex and naturalistic tasks [30] (i.e., moving from the simple task of two-button press to a carefully designed driving simulation task, to a real-world driving experiment).

- Procedure: The experiment was conducted while driving a simple rectangular course in an empty parking lot, with cues given to the driver in three different conditions. A set of random sequences of cues, to brake or accelerate, were presented to the driver (1) visually, (2) by audio, or (3) by using both audio and visual simultaneously. These sequences of Stop-and-Go cues were designed to be similar to a recent study on sequential effects in driving [30]. The driver responded as soon as possible by tapping the brake or acceleration pedal accordingly.

- Database: Experimental data was recorded on 12 subjects, of various nationalities, genders, and ages that range from their 20s to their 50s. All have a valid driver's license and ranged in experience from novice to decades of experience. Each subject did three runs, one under each of the three cueing conditions. A run includes 128 trials, where each trial includes one stop or go cue with the relevant response. Data collected in a synchronous manner include driver foot video, vehicle dynamics from CAN-bus, and information about the presented Stop-and-Go cues.

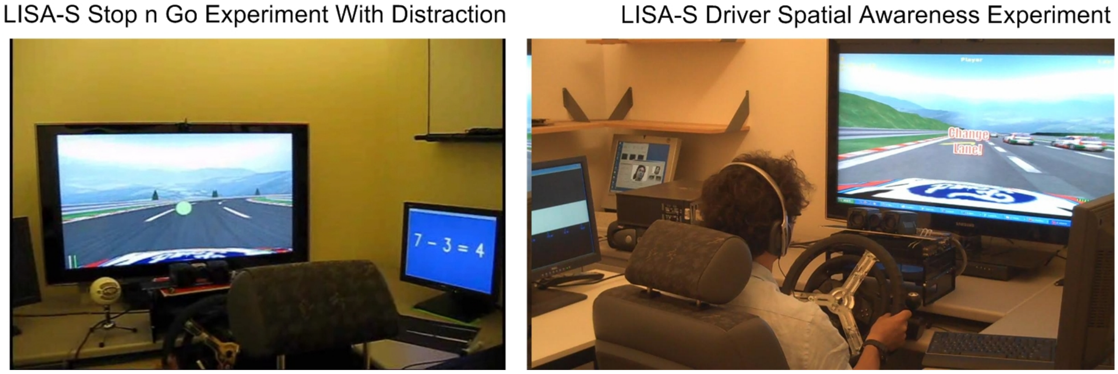

LISA-S Stop n Go Experiment With Distraction
7 − 3 = 4
LISA-S Driver Spatial Awareness Experiment
Change Lane!

Figure 5.6: LISA-S simulation experiments. Left - Stop-and-Go experiment with distraction. Right - Driver spatial awareness experiment.

5.2.2 Stop-and-Go Experiment with Distractions in LISA-S

This experiment is similar to the stop-and-go experiment in LISA-P with the additional cases of distraction and conflict between audio and visual cues (which are unsafe to do in real-world driving).

- Procedure: The simulation track for this experiment is a curvy, urban track compared to the parking lot track in LISA-P experiment. Subjects will respond to the Stop-and-Go cues similar to the real-world experiment. In the second half of the experiment, subjects will have an additional task (distraction) to respond 'YES' or 'NO' verbally based on the belief that the equations displayed on the side monitor are correct or incorrect. Subjects were told that a background color change on the side monitor would indicate a change in equation (blue to yellow, yellow to blue). We also add runs in which there is a conflict between audio and visual cues (e.g., audio cue is 'STOP' but visual cue shows 'GO').

- Database: We have captured data of 26 licensed participants (13 female, 13 male). The range of driving expertise is from 1 to 30 years. On average, our participants have driving experience of more than 6 years. Each subject does 6 runs V, A, AV, VD, AD, and AVD (each includes 128 trials) where V means Visual only (there are only visual cues), A-Audio only, AV-Audiovisual, VD-Visual only with distraction, AD-Audio only with distraction, AVD-Audiovisual with distraction. We also have 3 subjects doing two additional runs AVC-Audiovisual with a conflict between audio and visual cues and AVDC-Audiovisual with distraction and conflict. Data collected in a synchronous manner includes driver head and foot videos, FaceLAB head/eye tracker's output, audio input of driver's verbal responses, skin conductance sensor, information about the presented Stop-and-Go cues, and information about the ego vehicle and driving context from the simulator.

5.2.3 Driver Spatial Awareness Experiment in LISA-S

In this experiment, the objective is to study the effect of audio and visual cues in giving the driver a spatial awareness of the surrounding driving environment (related to the type of lateral critical scenarios).

- Procedure: At the beginning of the experiment, participants are introduced to the layout of the three-lane road track and the three other vehicles they will be driving with. Participants are told that their objective is to drive as fast as they can in a safe, comfortable manner without any collisions with the other three vehicles. They are instructed that when a "Change Lanes" message appears in the middle of the screen, whenever it is safe, they must change lanes. If a honking alert starts to occur due to the vehicle behind getting too close, it will be best to change lanes. The side monitors may be able to help as they will display oncoming traffic in your immediate left or right lane. When audio cues are enabled, a beeping alert will increase in volume as the nearby cars gets close. Lastly, red markings on the road entail a lane closure whereas arrows pointing in a certain direction will entail a lane entrance.
- Database: We have captured data of 40 licensed participants. The range of driving expertise is from 1 to 30 years. Each subject will do 2 runs either "1 run with visual cue only (side monitors), 1 run with audio (beeping alert) and visual cues" or "both with visual cues only." Each run is about 10 minutes. Data collected in a synchronous manner include driver head and foot videos, FaceLAB head/eye tracker's output, information about the presented cues, and information about the ego vehicle and driving context from the simulator.

5.3 Sequential Dependencies in Driving

In many naturalistic tasks, it is critically important for an individual to respond quickly to a sequence of cues in a rapidly changing environment. For example, drivers on highways worldwide are increasingly getting stuck in stop-and-go traffic [14], and are forced to engage in a sequence of braking and accelerating actions as dictated by cues from other vehicles. These situations sometimes take a turn for the worse: over 3,500 vehicles were involved in fatal rear-end collisions on highways in the U.S. in 2008 [100]. The effect of recent experience on current behavior has been studied extensively in simple laboratory tasks like pressing two buttons. Our objective is to explore the nature of sequential effects in the more naturalistic setting of automobile driving and how they could affect the design of driver assistance systems.

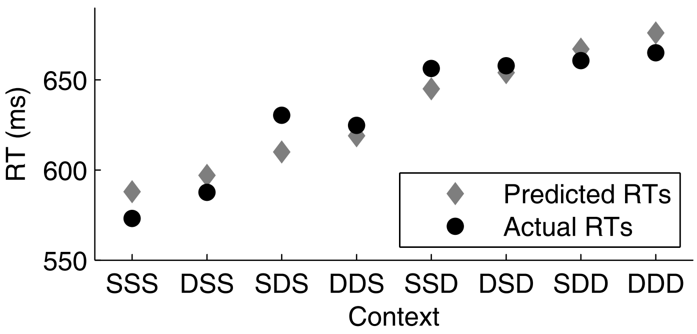


Figure 5.7: Mean overall reaction time (ORT) as a function of trial context (circles) and fit of first-order model with exponentially decaying memory (diamonds).

5.3.1 Experimental Analysis

Our research study of sequential dependencies in driving was done on a subset of the stop-and-go experiment in LISA-S (Section 5.2.2) with visual cues only, in which participants navigated a vehicle in a driving simulator and were occasionally required to brake or accelerate, as indicated by a traffic-light style red or green cue on the windshield. The analysis demonstrates the existence of sequential dependencies in both pedal-press latencies and errors.

Each trial in this analysis is characterized by (1) the target response (accelerate versus brake) and (2) the context arising from the three most recent trials. The context is specified in terms of whether the stimulus (or target response, the two are confounded in the present study) on the current trial, trial n, is the same as (S) or different than (D), the stimulus on trials n3, n2, and n1. For example, the context SDD represents a sequence in which trial n3 is the same as trial n, but trials n2 and n1 are different, which would correspond to the cue sequence red-green-green-red or green-red-red-green (ordering is always with the oldest first).

We present an analysis of response times (RTs), which is determined as the interval between when a cue (stop or go) appears and when the corresponding pedal is pressed. The mean RTs across subjects for each of three-back context are shown in Figure 5.7 (cirles). The contexts are ordered by similarity to the current trial, with recent trials being same response on the left and different response on the right. Similarity has a robust effect on RT. For the most extreme context contrast—SSS versus DDD—there is a reliable difference of ∼100 ms.

Theoretical accounts of sequential effects in two-alternative forced-choice tasks [21, 165] have obtained exquisite fits to data—accounting for over 98% of the variance in mean RT conditioned on the recent context—by assuming two distinct, additive mechanisms. Following the terminology of Wilder et al. [165], we refer to these effects as first and second-order. First-order effects depend on the recent history of stimulus/response identities, i.e., the S-D context. Second-order effects depend on the history of repetitions (R) and alternations (A) of the stimulus/response. For example, the trial sequence red-green-red-green corresponds to the first-order sequence DSD and the second-order sequence AAA; red-red-red-green corresponds to the first-order sequence DDD and the second-order sequence RRA. Although the first- and second-order sequences are in one-to-one correspondence, they can be dissociated in terms of the priming that they predict. For example, the sequence red-green-red-green produces first-order priming for green because two recent trials have been green, but strong second-order priming for red because the three recent trials have been alternations. In our experiment, the first-order model does a good job of fitting the patterns of RTs (Figure 5.7) while the second-order model obtains a poor fit. However, one must be cautious in favoring the combined model because the improvement could be due only to combined model's additional degrees of freedom.

5.3.2 Implications for Driver Assistance Systems

Consistent with previous studies in sequential effects with simple laboratory tasks, we find that recent events can result in response delays of nearly 100 ms, or 16%. Because sequential dependencies are found in complex laboratory conditions that approximate real-world driving, it seems altogether plausible that they will be observed in situations such as stop-and-go traffic, or a series of traffic lights in an urban environment.

At 65mph, a delay of 100 ms corresponds to an increased stopping distance of nearly three meters, which could be the difference between running into a leading vehicle and stopping safely. Even smaller differences in response times have been shown to have severe consequences in both the probability and severity of vehicle crashes [33].

Beyond delays in responding, we also found that recent events can affect the likelihood of a pedal error, which will be discussed in more detail in the following section. This context-dependent pattern of driver response delays and errors could potentially be exploited in a holistic Advanced Driver Assistance System, or ADAS [28, 149]. Given a particular observed history of pedal presses and familiarity with a particular driver, the ADAS could predict real-time performance. These predictions could be used to provide additional or earlier alerts to drivers in situations where delays/errors were likely. In more critical circumstances, for example, in stop-and-go traffic when the recent history of maneuvers forebodes an error or unacceptably high response time, the vehicle could take action to reduce impending accident severity by priming the brake system. In less critical situations, if the upcoming context is ripe for higher response times (e.g., more than 600 ms), an urgent alert could help to refocus the driver to the potentially appropriate response. An adaptive intelligent ADAS design could thus utilize sequential context to help the driver and mitigate dangerous or uncomfortable circumstances.

5.4 Understanding Pedal Errors and Audio-visual Stimuli

In this section, we discuss some initial analysis on our multimodal driving experiments, which provide insight into the effect of audio and visual cues on driver behavior such as the reaction time, movement time, and number of pedal errors.

5.4.1 Pedal Error Analysis

We will begin with discussion on some analysis on pedal errors based on the databases from stop-and-go experiment in LISA-P (Section 5.2.1) and stop-and-go experiment with distraction in LISA-S (Section 5.2.2). The term *pedal errors* here refers to the phenomenon when the driver is supposed to press a particular pedal (i.e., brake or acceleration) but s/he mistakenly presses the wrong pedal *(pedal misapplication)* or does not press any pedal at all *(pedal miss)* [138]. Typically, incidents related to pedal errors are reported based on driver surveys. Therefore, we want to understand more about the causes of pedal errors as well as how to predict and mitigate pedal errors. Some related research studies on pedal errors [37] and brake reaction time [81] with driving simulations showed that there were some relations between these events and driver's age and gender. In our experiments, we are also interested in other possible causes of pedal errors including the following:

- Driver workload: Analyze the effects of asking the driver to do additional tasks besides driving, e.g., verbally answering questions on the side monitor as in our stop-and-go with distraction experiment.
- Cue modality: Whether it could make a difference when the driver was stimulated to stop or accelerate by *visual*, *audio*, or both *audio and visual* cues.
- Sequential effect: Analyze the influence of one incidental experience on subsequent experience when individuals perform a series of tasks (i.e., the stop-and-go sequences in our experiment) [30].

Figure 5.8 shows the total number of pedal misapplications and pedal misses in different experiment configuration. Comparing "LISA-S w/o distraction" columns and "LISA-S w/ distraction" columns, we see that adding more workload (distraction) to the driver increases the number of pedal errors considerably. With regard to the cue modality, several subjects reported after the experiments that they felt more comfortable with audio or audio-visual stimuli than visual stimuli only. The reason could be that the driving task already puts significant loads on the driver's visual system. However, in the "LISA-P" and "LISA-S w/o distraction" columns, we see that though the driver feels more comfortable with audio and audio-visual stimuli, the number of errors does not seem to decrease. Looking at the "LISA-S w/ distraction" columns, using audio cue only seems to create more pedal errors. This may be because the distraction

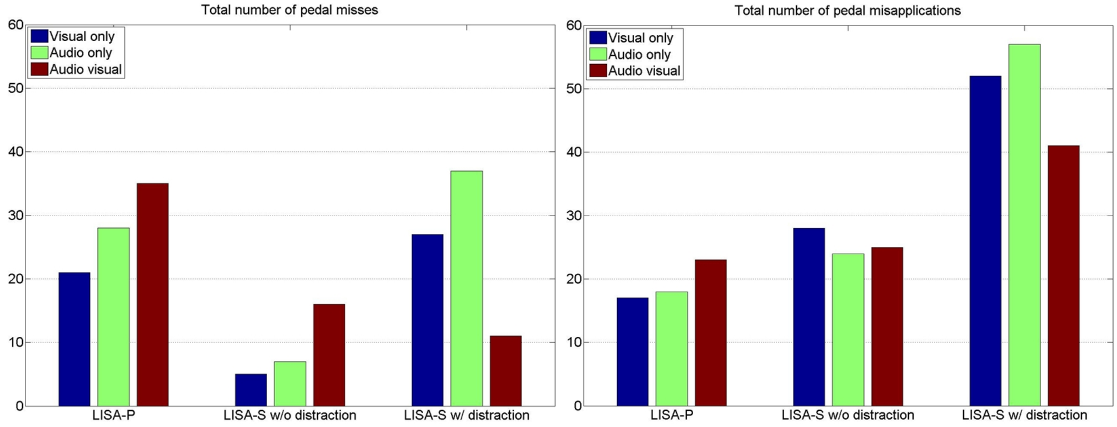


Figure 5.8: Analysis of pedal errors (misses and misapplications) in different experiment configurations.

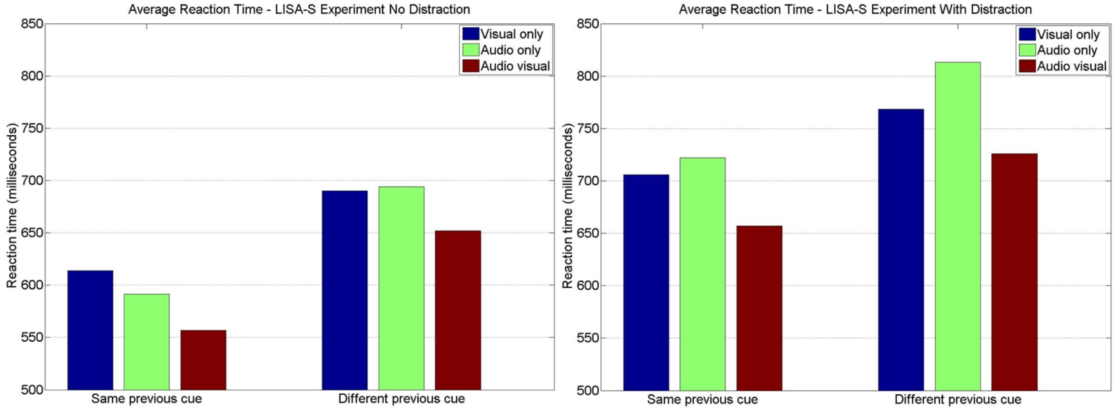


Figure 5.9: Average driver reaction time (time duration from the appearance of stop or go cue to the press of a corresponding pedal) in different experiment configurations.

task in our experiment requires the driver to verbally answer the question which may affect their ability in hearing the cue. However, it should be noted that when there are distractions, using multimodal (audio-visual) cue seems to obviously reduce the number of pedal errors. Another interesting observation is that in case of real-world LISA-P experiment, there seems to be more pedal misses than pedal misapplications while in the simulation experiment, it is the other direction. It seems that there are more safety concerns in the real-world driving so the subjects tend to be more careful in not doing the wrong thing.

Figure 5.9 shows the average driver reaction time in different experiment configuration. Here the reaction time is computed as the time duration from the appearance of stop or go cue to the press of a corresponding pedal. We can obviously see the sequential effect which causes a slower response when the previous cue is different compared to the case when the previous cue is the same. The additional distraction also causes a considerable increase in reaction time. We should also note that using multimodal (audio-visual) cue seems to consistently reduce the reaction time.

With regard to how to mitigate pedal errors, our approach for driver foot behavior modeling and prediction has been applied to the stop-and-go dataset from LISA-P (Section 5.2.1) [137]. The experimental results showed that a major portion of the pedal presses can be precisely predicted before they actually happen (e.g., recall rate of ∼74% at 133ms before the actual press). Among those, the pedal misapplications were all predicted correctly at ∼200ms in average before the actual press. This indicates the potential of using the proposed approach in predicting and mitigating pedal errors in real-world driving. For example, we can increase the pedal resistance when a misapplication is predicted. As reported in [2], haptic feedback is generally faster than visual feedback (∼50ms compared to 200 - 500ms) and involves less cognition.

5.4.2 Response Time Analysis

Based on the Stop-and-Go experiment in LISA-S (Section 5.2.2), we analyze the response time of pedal presses under different cue conditions (V: Visual only, A: Audio only, AV: Audio-visual) as well as the effect of distraction. Some main observations, as shown in Figure 5.10 and Figure 5.11, are the following:

- Response to audio-visual cue is faster than to audio or visual only. Though it may not be a real surprise, quantitative analysis is the contribution.

- Anecdotally, we see some interaction between audio and video cue, i.e., when the previous cue is the same, drivers response to visual cue faster than audio cue. However, when the previous cue is different, drivers response to audio cue faster. More experiments are needed to come up with a general conclusion about this.

- Adding distraction significantly increases the response time.

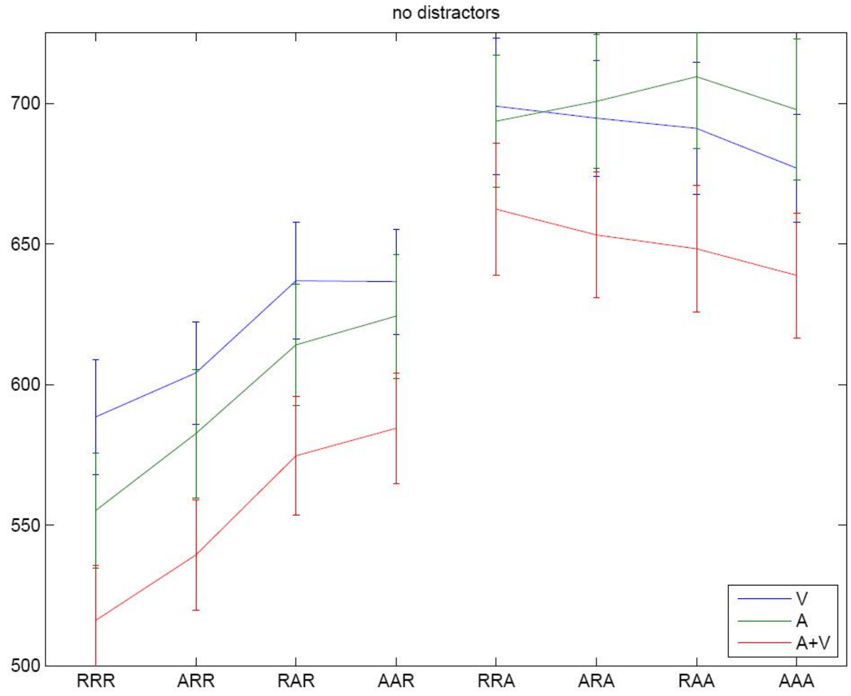


Figure 5.10: Sequential effects in pedal press response

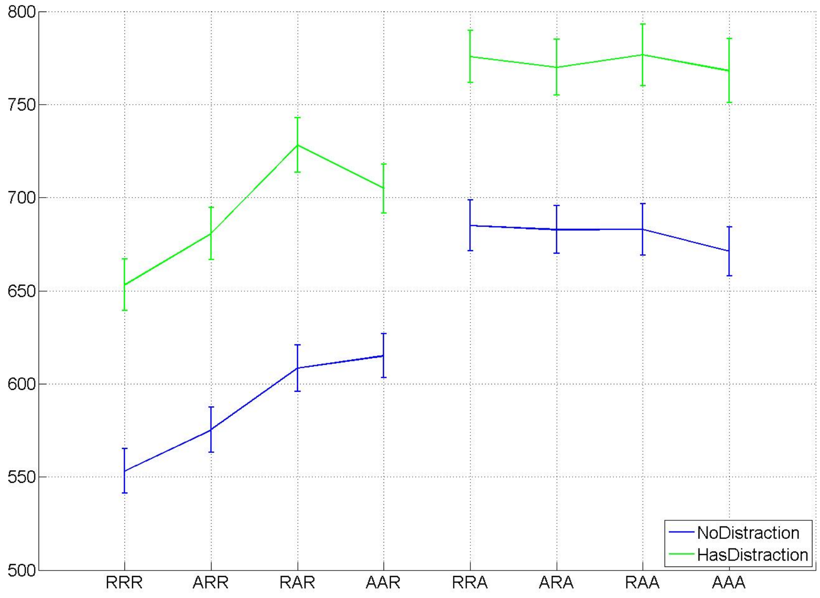


Figure 5.11: Effect of distraction in pedal press response.

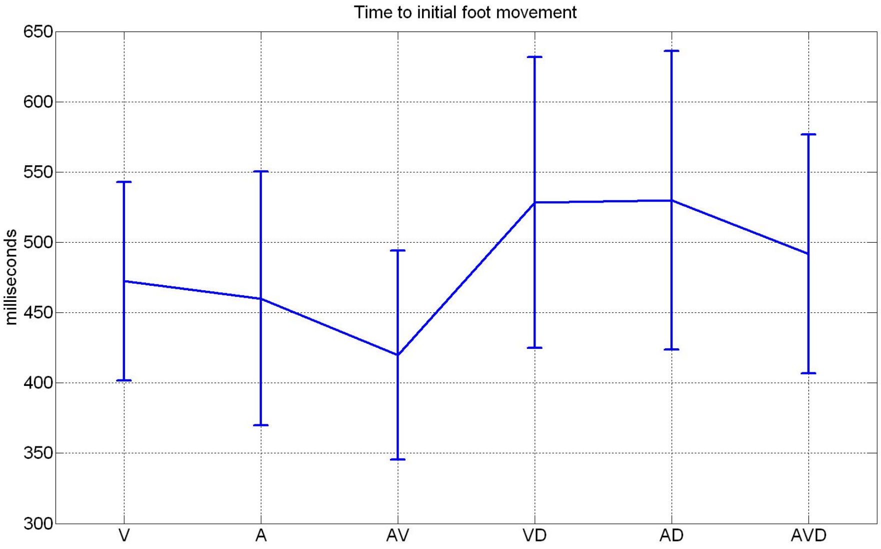


Figure 5.12: Time to initial foot movement (ms).

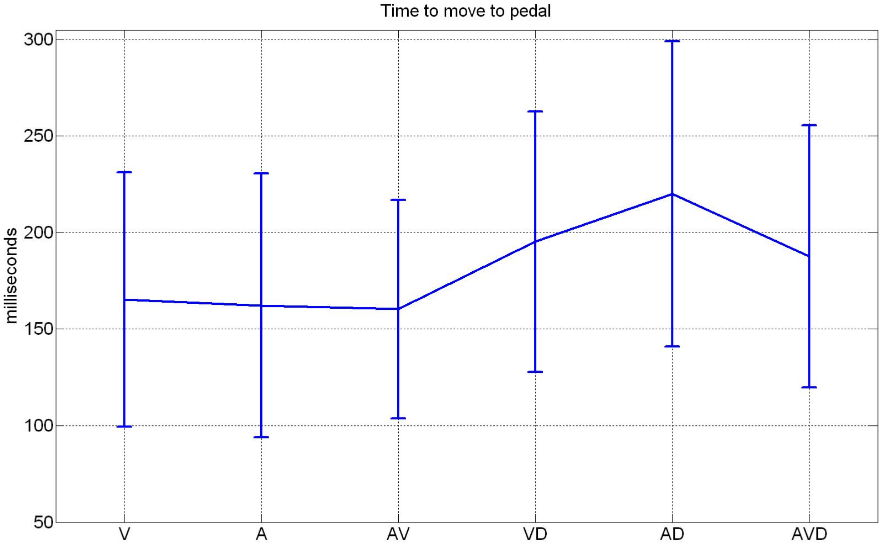


Figure 5.13: Time to move to pedal (ms).

5.4.3 Vision-based Foot Movement Time Analysis

Compared to pedal presses, vision-based foot movements could provide more useful information for behavior analysis. For a more detailed analysis, we break the total "time to pedal press" into "time to initial foot movement" and "time to move to pedal". Figure 5.12 shows the mean and standard deviation of "time to initial foot movement" under each cue condition. We still see a similar pattern as the analysis of pedal presses such that drivers response faster to audio-visual cue than visual or audio cue only, and distraction obviously slows down the responses. Figure 5.13 shows the mean and standard deviation of "time to move to pedal." We see that when there is no distraction, cue modality does not seem to have an effect on "time to move to pedal." However, when there are distractions, the "time to move to pedal" increases a little bit under audio-only condition. This might be because the type of distraction in this experiment requires the driver to answer verbally.

5.5 Acknowledgments

Chapter 5 is based on material that is published in the IEEE Intelligent Vehicles Symposium (2011) and IEEE International Conference on Intelligent Transportation Systems (2012) both by Cuong Tran, Anup Doshi, and Mohan M. Trivedi. The dissertation author was the primary investigator and author of these papers. Section 5.3 is also based on material published in the Cognitive Science (2012) by Anup Doshi, Cuong Tran, Matthew H. Wilder, Michael C. Mozer, and Mohan M. Trivedi, in which the dissertation author was a major author.

Chapter 6

Conclusions

Automatic perception of human posture and activity from vision input is an important research area with many potential applications. Though there has been a tremendous amount of related research studies and technology developments, it is still an open research area. More research efforts are needed to improve the accuracy, robustness, and efficiency of those technologies and promote their usability more and more in our daily life. This dissertation focuses on developing systems that look at humans at multiple levels of detail to better understand natural human activities. We also put an emphasis on achieving the robustness and efficiency (real-time performance) which are required for interactive applications such as intelligent driver assistance systems, gesture-based interactive games, and smart rooms. Novel approaches for human pose modeling and tracking at different levels are presented in Chapter 3 including (1) an integrated framework with automatic initialization for body and hand pose modeling and tracking from voxel data and (2) a real-time upper body pose tracker in 3-D using extremities movement observation XMOB. Novel approaches for multilevel human gesture analysis for interactivity are presented in Chapter 4 including: (1) a system based on XMOB that does both real-time upper body pose tracking and gesture recognition based on pose tracking output, (2) a system combining upper body and head tracking information for driver assistance "Keeping hands on the wheel and eyes on the road," and (3) a framework for driver foot and head behavior analysis based on optical flow tracking.

In addition to common computer vision challenges such as the occlusion issue, background clutter, variable lighting condition and human appearance, there are some other research issues in developing systems with the emphasis on multilevel posture and activity analysis for interactive applications. For more convenient real-world deployments, it is desirable to develop some efficient analysis frameworks which can be easily adapted to different body levels instead of developing totally separate approaches for each level. In this regard, our proposed framework based on optical flow tracking in Section 4.3 was applied well for both driver foot and head behavior analysis. Our proposed integrated framework with automatic initialization (Section 3.1) also showed good results for both body and hand pose modeling and tracking. In this work, two existing approaches are integrated into a more powerful system in which we improved the KC-GMM pose tracking method with an automatic initialization as well as reduce the sensitivity to voxel noise of the LE-based voxel segmentation method. Section 3.3 shows our initial step in modeling and tracking a full model of different body levels in which we still track each body level separately and then combine their outputs using calibration parameters. A more interesting problem is how to coestimate and analyze human pose and activity at different levels simultaneously in which the information from different levels can support each other and help to improve the overall performance. This is still an open and challenging question.

Another research issue is how to deal with the typical trade-offs in developing generic versus application-specific approaches for robustness and efficiency which are particularly important for interactive applications. A common principle that we follow is try to make "everything as simple as possible, but not simpler" *(Albert Einstein)*. Focusing on situations where the arms carry the most influential information of body motion (e.g., in meeting room, teleconference, driver assistance situations), our proposed XMOB upper body tracker using extrimity movement (Section 3.2) showed several advantages. With regard to robustness, the underlying idea is to use the easier parts to track: the head and hands, which have less occlusion and typically are well defined with skin color, to help the tracking of harder inner parts (i.e., shoulders and elbows). By breaking the upper body tracking problem into two subproblems (first track the extremal parts then infer the whole upper body pose from head and hand movements as an inverse kinematics problem), the complexity is also reduced considerably to achieve real-time performance. Furthermore, XMOB can recover from failures, and it will work as long as the head and hands are observable (it does not matter if the user wears very loose clothes or the clothes' colors are mixed with the background which could be a difficult case for other approaches). The downside of possible ambiguities due to the kinematic redundancy of upper body model was resolved using more constrained body motion space in the mentioned situations.

Our optical flow-based framework for driver foot and head behavior analsyis (Section 4.3) was also developed to specifically work with driving situations where there are more constraints on the type of body motion. To our knowledge, our framework is the very first vision-based system for driver foot behavior modeling and prediction. The experimental evaluation showed good results in both estimation of driver foot behavior states {Neutral, TowardsBrake, TowardsAccel, BrakeEngage, AccelEngage, ReleaseBrake, ReleaseAccel} as well as prediction of a pedal press at $\sim$ 133ms before it actually happens. This time advantage could be important in developing driver assistance systems to mitigate pedal misapplication (unintended acceleration) incidents, for example. A similar optical flow-based framework was also applied well to monitor driver head activity. Our simulation experiment indicates that if we are only concerned about some common states of driver head {LookingStraight, LookingLeft, LookingRight}, the proposed optical flow-based framework provides results comparable to a commercial head tracking systems which is much more expensive to deploy.

In interactive applications, the user and system work in collaboration. Therefore, the system might also expect some supporting feedback from the user which could help to ease the difficulties and improve system performance. In some of our works including the integrated framework with automatic initialization for body and hand pose modeling and tracking (Section 3.1) and the XMOB upper body tracker (Section 3.2), users are also asked to start with some specific pose and gesture to help the systems in the initialization step. However, more general frameworks to incorporate user feedback into different stages of the system of human pose and activity analysis are desirable and should be studied in the future.

One main targeted application domain of this dissertation is to develop intelligent driver assistance systems. In Section 4.2, we developed driver assistance system for “Keeping hands on the wheel and eyes on the road” by combining information from driver head and hand activities. In Chapter 5, we also discuss the issue of building adequate multimodal testbeds and experiments as well as utilizing the ability to analyze driver pose and activity at different levels of detail in developing efficient driver assistance systems. We present our development of multimodal driving testbeds as well as several joint audio-visual experiments and databases that we have developed for studying traffic accident prevention. These experiments focus on two typical types of traffic accident scenarios which are longitudinal accidents (i.e., head-on and rear-end collisions) and lateral accidents (i.e., side collisions). Based on those multimodal experiments and databases, we present our analysis which provides insight into the effect of sequential dependencies on driving (different recent pedal press events can result in response delays of nearly 100ms) as well as some initial quantitative analysis of the effect of audio and visual cues on driver behavior such as the reaction time, movement time, and number of pedal errors. Though more works still remain to be done, these studies show high promise in implementation and design of driver assistance systems for improving active safety and driver experience on the road.

Bibliography

[1] MuHAVi dataset instructions at http://dipersec.king. ac.uk/MuHAVi-MAS/.

[2] D.A. Abbink, E.R. Boer, and M. Mulder. Motivation for Continuous Haptic Gas Pedal Feedback to Support Car Following. *IEEE Intelligent Vehicles Symposium*, 2008.

[3] M. Ahmad and S.-W. Lee. Hmm-based human action recognition using multiview image sequences. In *ICPR*, 2006.

[4] A.A. Alonso, R.D Rosa, L.D. Val, M.I. Jimenez, and S. Franco. A robot controlled by blinking for ambient assisted living. *Proceedings of the 10th International Work-Conference on Artificial Neural Networks: Part II: Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*, 2009.

[5] M. Ankerst, G. Kastenmüller, H.-P. Kriegel, and T. Seidl. 3d shape histograms for similarity search and classification in spatial databases. In *SSD*, 1999.

[6] J. Assfalg, M. Bertini, C. Colombo, A.D. Bimbo, and W. Nunziati. Semantic annotation of soccer videos: automatic highlights identification. *CVIU*, 92(2-3), 2003.

[7] M.S. Bartlett, G. Littlewort, M.G. Frank, C. Lainscsek, I. Fasel, and J.R. Movellan. Automatic Recognition of Facial Actions in Spontaneous Expressions. *Journal of Multimedia*, 1(6), 2005.

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

[9] M. Benali-Khoudja, M. Hafez, J.M. Alexandre, and A. Kheddar. Tactile interfaces: a state-of-the-art survey. *International Symposium on Robotics*, 2004.

[10] O. Bernier, P. Cheung-Mon-Chana, and A. Bougueta. Fast nonparametric belief propagation for real-time stereo articulated body tracking. *Computer Vision and Image Understanding*, 113(1):29–47, 2009.

[11] F. Caillette, A. Galata, and T. Howard. Real-time 3-D Human Body Tracking Using Learnt Models of Behaviour. *Computer Vision and Image Understanding*, 109:112–125, 2008.

[12] C. Canton-Ferrer, J.R. Casas, and M. Pardás. Human model and motion based 3d action recognition in multiple view scenarios. In *EUSIPCO*, 2006.

[13] C.S. Chan, H. Liu, and D.J. Brown. Human Arm-Motion Classification Using Qualitative Normalized Templates. *Lecture Notes in Artificial Intelligence*, pages 639–646, 2006.

[14] Anita Chang. China's massive traffic jam could last for weeks. Associated Press, August 24 2010.

[15] Shinko Y. Cheng and Mohan M. Trivedi. Multimodal voxelization and kinematically constrained gaussian mixture model for full hand pose estimation: An integrated systems approach. In *ICVS*, 2006.

[16] S.Y. Cheng and M.M. Trivedi. Turn-Intent Analysis Using Body Pose for Intelligent Driver Assistance. *IEEE Pervasive Computing*, 5(4):28–37, 2006.

[17] S.Y. Cheng and M.M. Trivedi. Articulated Human Body Pose Inference from Voxel Data Using a Kinematically Constrained Gaussian Mixture Model. *CVPR EHuM2: 2nd Workshop on Evaluation of Articulated Human Motion and Pose Estimation*, 2007.

[18] S. Cherla, K. Kulkarni, A. Kale, and V. Ramasubramanian. Towards fast, view-invariant human action recognition. In *CVPR Workshops*, 2008.

[19] G. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. *IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.

[20] G. Cheung and T. Kanade. A real-time system for robust 3d voxel reconstruction of human motions. In *CVPR*, 2000.

[21] R.Y. Cho, L.E. Nystrom, E.T. Brown, A.D. Jones, T.S. Braver, P.J. Holmes, and J.D. Cohen. Mechanisms underlying dependencies of performance on stimulus history in a two-alternative forced-choice task. *Cognitive Affective Behavior Neuroscience*, 2(4):283–299, 2002.

[22] I. Choi and C. Ricci. Foot-Mounted Gesture Detection and its Application in Virtual Environments. *IEEE Intl. Conf. on Systems, Man, and Cybernetics*, 1997.

[23] I. Cohen and H. Li. Inference of human postures by classification of 3d human body shape. In *AMFG*, 2003.

[24] S. Corazza, L. Mundermann, E. Gambaretto, G. Ferrigno, and T. Andriacchi. Markerless motion capture through visual hull, articulated ICP and subject specific model generation. *Intl. Journal of Computer Vision*, 87(1-2), 2010.

[25] A. Datta, Y. Sheikh, and T. Kanade. Linear Motion Estimation for Systems of Articulated Planes. *IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.

[26] D. DeCarlo and D. Metaxas. Optical Flow Constraints on Deformable Models with Applications to Face Tracking. *Intl. Journal of Computer Vision*, 38:99–127, 2000.

[27] Q. Delamarre and O. Faugeras. 3d articulated models and multiview tracking with physical forces. *CVIU*, 81(3):328–357, 2001.

[28] A. Doshi, S.Y. Cheng, and M.M. Trivedi. A Novel, Active Heads-up Display for Driver Assistance. *IEEE Trans. on Systems, Man, and Cybernetics - Part B*, 39, 2009.

[29] A. Doshi, B. Morris, and M.M. Trivedi. On-Road Prediction of Driver's Intent with Multimodal Sensory Cues. *IEEE Pervasive, Special Issue on Automotive Pervasive Computing*, 2011.

[30] A. Doshi, C. Tran, M. Wilder, M.C. Mozer, and M.M. Trivedi. Sequential Dependencies in Driving. *Cognitive Science*, to appear 2012.

[31] A. Doshi and M.M. Trivedi. Satellite Imagery Based Robust, Adaptive Background Models and Shadow Suppression. *Signal, Image, and Video Processing (SIViP) Journal*, 1(2):119–132, 2007.

[32] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, 108(1-2), 2007.

[33] L. Evans. *Traffic Safety and the Driver.* Van Nostrand Reinhold, New York, 1991.

[34] A. Farhadi and M.K. Tabrizi. Learning to recognize activities from the wrong view point. In *ECCV*, 2008.

[35] V. Ferrari, M. Jimnez, and A. Zisserman. Progressive Search Space Reduction for Human Pose Estimation. *IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.

[36] Preben Fihl and Thomas B. Moeslund. Invariant gait continuum based on the duty-factor. *SIViP*, 3(4):391–402, 2008.

[37] B. Freund, L.A. Colgrovea, D. Petrakosa, and R. McLeod. In my car the brake is on the right: Pedal errors among older drivers. *Accident Analysis and Prevention*, 2008.

[38] M. Gales and S. Young. The application of hidden Markov models in speech recognition. *Foundations and Trends in Signal Processing*, 1(3), 2008.

[39] J. Gall, B. Rosenhahn, T. Brox, and H. Seidel. Optimization and Filtering for Human Motion Capture: A Multi-Layer Framework. *Intl. Journal of Computer Vision*, 87(1-2):75–92, 2010.

[40] S. S. Ge, Y. Yang, and T. H. Lee. Hand gesture recognition and tracking based on distributed locally linear embedding. *Image and Vision Computing*, 26(12):1607–1620, 2008.

[41] J. Geigel and M. Schweppe. Motion capture for realtime control of virtual actors in live, distributed, theatrical performances. In *FG*, 2011.

[42] N. Gkalelis, H. Kim, A. Hilton, N. Nikolaidis, and I. Pitas. The i3dpost multi-view and 3d human action/interaction database. In *CVMP*, 2009.

[43] N. Gkalelis, N. Nikolaidis, and I. Pitas. View indepedent human movement recognition from multi-view video exploiting a circular invariant posture representation. In *ICME*, 2009.

[44] G. Gomez and E. Morales. Automatic Feature Construction and a Simple Rule Induction Algorithm for Skin Detection. *ICML Workshop on Machine Learning in Computer Vision*, pages 31–38, 2002.

[45] R. Gross and J. Shi. The cmu motion of body (mobo) database. In *Techical Report*, 2001.

[46] H. Gunes and M. Piccardi. Automatic Temporal Segment Detection and Affect Recognition From Face and Body Display. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 38(5), 2008.

[47] A. Haq, I. Gondal, and M. Murshed. On dynamic scene geometry for view-invariant action matching. In *CVPR*, 2011.

[48] J.R. Healey and S.S. Carty. Driver error found in some Toyota acceleration cases. *USA Today*, 2010.

[49] M. Hofmann and D.M. Gavrila. Multi-view 3D Human Pose Estimation in Complex Environment. *Intl. Journal of Computer Vision*, 2011.

[50] M.B. Holte, T.B. Moeslund, and P. Fihl. View-invariant gesture recognition using 3D optical flow and harmonic motion context. *Computer Vision and Image Understanding*, 114:1353–1361, 2010.

[51] M.B. Holte, T.B. Moeslund, N. Nikolaidis, and I. Pitas. 3d human action recognition for multi-view camera systems. In *3DIMPVT*, 2011.

[52] M.B. Holte, C. Tran, M.M Trivedi, and T.B. Moeslund. Human Pose Estimation and Activity Recognition From Multi-View Videos: Comparative Explorations of Recent Developments. *IEEE Journal of Selected Topics in Signal Processing*, 6(5), 2012.

[53] T. Horprasert, D. Harwood, and L.S. Davis. A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection. *IEEE Proceedings ICCV Frame-Rate Workshop*, 1999.

[54] K. S. Huang and M. M. Trivedi. 3D Shape Context Based Gesture Analysis Integrated with Tracking using Omni Video Array. *IEEE Workshop on Vision for Human Computer Interaction (V4HCI), in conjunction with IEEE CVPR Conference*, 2005.

[55] K.S. Huang and M.M. Trivedi. Robust Real-Time Detection, Tracking, and Pose Estimation of Faces in Video Streams. *Intl. Conf. on Pattern Recognition*, 2004.

[56] P. Huang and A. Hilton. Shape-colour histograms for matching 3d video sequences. In *3DIM*, 2009.

[57] P. Huang, A. Hilton, and J. Starck. Shape similarity for 3d video sequences of people. *IJCV*, 89:362–381, 2010.

[58] E. Hunter. Visual estimation of articulated motion using expectation-constrained maximization algorithm. In *PhD thesis, UCSD*, 1999.

[59] Z. Husz and A. Wallace. Evaluation of a hierarchical partitioned particle filter with action primitives. In *CVPR Workshops*, 2007.

[60] B.-W. Hwang, S. Kim, and S.-W. Lee. A fullbody gesture database for automatic gesture recognition. In *FG*, 2006. http://gesturedb.korea.ac.kr/.

[61] A. Iosifidis, N. Nikolaidis, and I. Pitas. Movement recognition exploiting multi-view information. In *MMSP*, 2010.

[62] A. Jaimes and N. Sebe. Multimodal Human Computer Interaction: A Survey. *Computer Vision and Image Understanding*, 108(1-2):116–134, 2007.

[63] X. Ji and H. Liu. Advances in view-invariant human motion analysis: A review. *Trans. Sys. Man Cyber Part C*, 40(1):13–24, 2010.

[64] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics*, 14(2):201–211, 1973.

[65] A.E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *PAMI*, 21(5):433–449, 1999.

[66] I.N. Junejo, E. Dexter, I. Laptev, and P. Pérez. Cross-view action recognition from temporal self-similarities. In *ECCV*, 2008.

[67] I.N. Junejo, E. Dexter, I. Laptev, and P. Pérez. View-independent action recognition from temporal self-similarities. *PAMI*, 33(1):172–185, 2011.

[68] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *SGP*, 2003.

[69] D. Kelly, J. McDonald, and C. Markham. A person independent system for recognition of hand postures used in sign language. *Pattern Recognition Letters*, 31(11):1359–1368, 2010.

[70] J. Kilner, J.-Y. Guillemaut, and A. Hilton. 3d action matching with key-pose detection. In *ICCV Workshops*, 2009.

[71] D. Knossow, R. Ronfard, and R. Horaud. Human motion tracking with a kinematic parameterization of extremal contours. *IJCV*, 79(3):247–269, 2008.

[72] M. Körtgen, M. Novotni, and R. Klein. 3d shape matching with 3d shape contexts. In *CESCG*, 2003.

[73] P. P. Kumar, P. Vadakkepat, and L. A. Poh. Hand Posture and Face Recognition Using A Fuzzy-Rough Approach. *Intl. Journal of Humanoid Robotics*, 7(3):331–356, 2010.

[74] H. Lee and J. H. Kim. An HMM-Based Threshold Model Approach for Gesture Recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(10), 1999.

[75] M.W. Lee and I. Cohen. Human Upper Body Pose Estimation in Static Images. *European Conf. on Computer Vision*, 2004.

[76] L. Li, X. Li, C. Cheng, C. Chen, G. Ke, D. Zeng, and W.T. Scherer. Research Collaboration and ITS Topic Evolution: 10 Years at T-ITS. *IEEE Transactions on Intelligent Transportation Systems*, 2010.

[77] R. Li, T.P. Tian, S. Sclaroff, and M.H. Yang. 3D human motion tracking with a coordinated mixture of factor analyzers. *Intl. Journal of Computer Vision*, 87(1-2), 2010.

[78] J. Liu, S. Ali, and M. Shah. Recognizing human actions using multiple features. In *CVPR*, 2008.

[79] J. Liu and M. Shah. Learning human actions via information maximization. In *CVPR*, 2008.

[80] Jingen Liu, Mubarak Shah, Benjamin Kuipers, and Silvio Savarese. Cross-view action recognition via view knowledge transfer. In *CVPR*, 2011.

[81] L. Livne and D. Shinar. Effects of Uncertainty, Transmission Type, Driver Age and Gender on Brake Reaction and Movement Time. *Journal of Safety Research*, 2002.

[82] B. Lucas and T. Kanade. An Iterative Image Registration Technique with An Application To Stereo Vision. *Proceedings of Imaging Understanding Workshop*, 1981.

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

[84] Pyry Matikainen, Padmanabhan Pillai, Lily Mummert, Rahul Sukthankar, and Martial Hebert. Prop-free pointing detection in dynamic cluttered environments. In *FG*, 2011.

[85] J. McCall, D. Wipf, M.M. Trivedi, and B. Rao. Lane Change Intent Analysis Using Robust Operators and Sparse Bayesian Learning. *IEEE Transactions on Intelligent Transportation Systems*, 2007.

[86] J.C. McCall and M.M. Trivedi. Driver Behavior and Situation Aware Brake Assistance for Intelligent Vehicles. *Proceedings of the IEEE*, 95(2):374–387, 2007.

[87] M. C. D. Mendonca. Estimation of height from the length of long bones in a portugese adult population. *American Journal of Physical Anthropology*, 2000.

[88] A.S. Micilotta, E. Ong, and R. Bowden. Real-time Upper Body Detection and 3D Pose Estimation in Monoscopic Images. *European Conf. on Computer Vision*, 2006.

[89] I. Mikic, M.M. Trivedi, E. Hunter, and P. Cosman. Human Body Model Acquisition and Tracking using Voxel Data. *Intl. Journal of Computer Vision*, 51(3), 2003.

[90] T. Moeslund, A. Hilton, and V. Krueger. A Survey on Advances in Vision-based Human Motion Capture and Analysis. *Computer Vision and Image Understanding*, 104(2), 2006.

[91] T.B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *CVIU*, 81(3):231–268, 2001.

[92] T.B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *CVIU*, 104(2-3):90–126, 2006.

[93] L. Muendermann, S. Corazza, A.M. Chaudhari, T.P. Andriacchi, A. Sundaresan, and R. Chellappa. Measuring human movement for biomechanical applications using markerless motion capture. In *Proceeding of SPIE Three-Dimensional Image Capture and Applications*, 2006.

[94] M. Mulder, J.J.A. Pauwelussen, M.M. van Paassen, M. Mulder, and D.A. Abbink. Active Deceleration Support in Car Following. *IEEE Trans. on Systems, Man, and Cybernetics - Part A*, 40(6), 2010.

[95] E. Murphy-Chutorian and M.M. Trivedi. 3d tracking and dynamic analysis of human head movements and attentional targets. In *IEEE/ACM Int'l. Conf. on Distributed Smart Cameras*, 2008.

[96] E. Murphy-Chutorian and M.M. Trivedi. Head pose estimation in computer vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):607–626, 2009.

[97] E. Murphy-Chutorian and M.M. Trivedi. Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness. *IEEE Transactions on Intelligent Transportation Systems*, 2010.

[98] E. Murphy-Chutorian and M.M. Trivedi. Head Pose Estimation and Augmented Reality Tracking: An Integrated System and Evaluation for Monitoring Driver Awareness. *IEEE Trans. on Intelligent Transportation Systems*, 2010.

[99] NHTSA. Traffic Safety Facts 2006 - A Compilation of Motor Vehicle Crash Data From the Fatality Analysis Reporting System and the General Estimates System. *Washington, DC: Nat. Center Stat. Anal., US Dept. Transp.*, 2006.

[100] NHTSA. National Highway Traffic Safety Administration - Fatal Analysis Reporting System (FARS). *available online - www-fars.nhtsa.dot.gov*, 2010.

[101] A. Oikonomopoulos and M. Pantic. Human Body Gesture Recognition using Adapted Auxiliary Particle Filtering. *IEEE Intl. Conf. on Advanced Video and Signal Based Surveillance*, 2007.

[102] World Health Organization. World Report on Road Traffic Injury Prevention: Summary. *Technical Report*, 2004.

[103] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Trans. Graph.*, 21:807–832, 2002.

[104] S. Park and T.B. Sheridan. Enhanced Human Machine Interface in Braking. *IEEE Trans. on Systems, Man, and Cybernetics - Part A*, 34(5), 2004.

[105] S. Park and M.M. Trivedi. Understanding Human Interactions with Track and Body Synergies (TBS) Captured from Multiple Views. *Computer Vision and Image Understanding*, 111(1):2–20, 2008.

[106] B. Paulson, D. Cummings, and T. Hammond. Object interaction detection using hand posture cues in an office setting. *Intl. Journal of Human-Computer Studies*, 69(1-2):19–29, 2011.

[107] S. Pehlivan and P. Duygulu. A new pose-based representation for recognizing actions from multiple cameras. *CVIU*, 115:140–151, 2011.

[108] A. Pentland and A. Liu. Modeling and Prediction of Human Behavior. *Neural Computation*, 11, 1999.

[109] M. Pierobon, M. Marcon, A. Sarti, and S. Tubaro. 3-d body posture tracking for human action template matching. In *ICASSP*, 2006.

[110] J. Pollard and E. D. Sussman. An Examination of Sudden Acceleration. *Report DOT HS 807367, NHTSA, U.S. Department of Transportation*, 1989.

[111] R. Poppe. Vision-based Human Motion Analysis: An Overview. *Computer Vision and Image Understanding*, 108:4–18, 2007.

[112] R. Poppe. A survey on vision-based human action recognition. *IVC*, 28(6):976–990, 2010.

[113] Ronald Poppe. Evaluating example-based pose estimation: Experiments on the humaneva sets. In *CVPR Workshops*, 2007.

[114] A.B. Postawa, M. Kleinsorge, J. Krueger, and G. Seliger. Automated image based recognition of manual work steps in the remanufacturing of alternators. *Advances in Sustainable Manufacturing*, 5:209–214, 2011.

[115] Y. Kogay and K. Kondoz and J. Kuffnery and J. Latombey. Planning motions with intentions. *SIGGRAPH*, 1994.

[116] L.R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2), 1989.

[117] J. Radmer and J. Krueger. Depth data-based capture of human movement for biomechanical application in clinical rehabilitation use. In *5th International Symposium on Health Informatics and Bioinformatics*, 2010.

[118] M. Ramsey. Toyota Rethinks Pedal Design. *The Wall Street Journal*, 2010.

[119] O. Rashid, A. Al-Hamadi, and B. Michaelis. A Framework for the Integration of Gesture and Posture Recognition Using HMM and SVM. *2009 IEEE Intl. Conf. on Intelligent Computing and Intelligent Systems*, 2009.

[120] K.K. Reddy, J. Liu, and M. Shah. Incremental action recognition using feature-tree. In *ICCV*, 2009.

[121] G. Rigoll, A. Kosmala, and S. Eickeler. High Performance Real-Time Gesture Recognition Using Hidden Markov Models. *Intl. Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction*, 1998.

[122] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau. Fall detection from human shape and motion history using video surveillance. In *21st International Conference on Advanced Information Networking and Applications Workshops*, 2007.

[123] M.M. Trivedi S. Park. Multi-person interaction and activity analysis: a synergistic track- and body-level analysis framework. *Machine Vision and Applications: Special Issue on Novel Concepts and Challenges for the Generation of Video Surveillance Systems*, 18(3-4):151–166, 2007.

[124] S. Shivappa, M. M. Trivedi, and B. D. Rao. Audio-visual Information Fusion In Human Computer Interfaces and Intelligent Environments: A Survey. *Proceedings of the IEEE*, 2010.

[125] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, 2011.

[126] L. Sigal and M. J. Black. Guest Editorial: State of the Art in Image and Video Based Human Pose and Motion Estimation. *Intl. Journal of Computer Vision*, 87(1):1–3, 2010.

[127] L. Sigal and M.J. Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. In *Techniacl Report*, 2006.

[128] G. Slabaugh, B. Culbertson, and T. Malzbender. A survey of methods for volumetric scene reconstruction for photographs. In *VG*, 2001.

[129] G. Slabaugh, R. Schafer, and M. Hans. Image based photo hulls. In *3DPVT*, 2002.

[130] J. Soechting and M. Flanders. Errors in pointing are due to approximations in sensorimotor transformations. *Journal of Neurophysiology*, 62(2):595–608, 1989.

[131] Y. Song, D. Demirdjian, and R. Davis. Multi-signal gesture recognition using temporal smoothing hidden conditional random fields. In *FG*, 2011.

[132] R. Souvenir and J. Babbs. Learning the viewpoint manifold for action recognition. In *CVPR*, 2008.

[133] J. Starck and A. Hilton. Surface capture for performance based animation. *IEEE Computer Graphics and Applications*, 27(3):21–31, 2007.

[134] A. Sundaresan and R. Chellappa. Model driven segmentation of articulating humans in Laplacian Eigenspace. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(10):1771–1785, 2008.

[135] Y. Tanaka, H. Kaneyuki, T. Tsuji, T. Miyazaki, K. Nishikawa, and T. Nouzawa. Mechanical and Perceptual Analyses of Human Foot Movements in Pedal Operation. *IEEE Intl. Conf. on Systems, Man, and Cybernetics*, 2009.

[136] T. Tangkuampien and D. Suter. Real-Time Human Pose Inference using Kernel Principal Component Pre-image Approximations. *British Machine Vision Conference*, 2006.

[137] C. Tran, A. Doshi, and M. M. Trivedi. Modeling and prediction of driver behavior by foot gesture analysis. *Computer Vision and Image Understanding*, 116(3):435–445, 2012.

[138] C. Tran, A. Doshi, and M.M. Trivedi. Pedal Errors Prediction by Driver Foot Gesture Analysis: A Vision-based Inquiry. *IEEE Intelligent Vehicles Symposium*, 2011.

[139] C. Tran and M. M. Trivedi. Driver Assistance for Keeping Hands on the Wheel and Eyes on the Road. *IEEE Intl. Conf. on Vehicular Electronics and Safety*, 2009.

[140] C. Tran and M. M. Trivedi. Introducing XMOB: Extremity Movement Observation Framework for Upper Body Pose Tracking in 3D. *Demo paper (2 pages), IEEE Intl. Symposium on Multimedia*, 2009.

[141] C. Tran and M.M. Trivedi. Hand modeling and tracking from voxel data: An integrated framework with automatic initialization. In *IEEE International Conference on Pattern Recognition*, 2008.

[142] C. Tran and M.M. Trivedi. Human body modeling and tracking using volumetric representation: Selected recent studies and possibilities for extensions. In *ACM workshops*, 2008.

[143] C. Tran and M.M. Trivedi. Driver assistance for 'keeping hands on the wheel and eyes on the road. In *IEEE International Conference on Vehicular Electronics and Safety*, 2009.

[144] D. Tran and A. Sorokin. Human activity recognition with metric learning. In *ECCV*, 2008.

[145] M.M. Trivedi. Human movement capture and analysis in intelligent environments. *Machine and Vision Applications*, 14(4):215–217, 2003.

[146] M.M. Trivedi and S.Y. Cheng. Holistic sensing and active displays for intelligent driver support systems. In *IEEE Computer Magazine*, 2007.

[147] M.M. Trivedi and S.Y. Cheng. Holistic Sensing and Active Displays for Intelligent Driver Support Systems. *IEEE Computer*, 2007.

[148] M.M. Trivedi, S.Y. Cheng, E. Childers, and S. Krotosky. Occupant posture analysis with stereo and thermal infrared video: Algorithms and experimental evaluation. *IEEE Transactions on Vehicular Technology, Special Issue on In-Vehicle Vision Systems*, 53(6), 2004.

[149] M.M. Trivedi, T. Gandhi, and J. McCall. Looking-In and Looking-Out of a Vehicle: Computer-Vision-Based Enhanced Vehicle Safety. *IEEE Trans. on Intelligent Transportation Systems*, pages 108–120, 2007.

[150] M.M. Trivedi, K.S. Huang, and I. Mikic. Dynamic context capture and distributed video arrays for intelligent spaces. *IEEE Trans. on Systems, Man and Cybernetics, Part A*, 35(1):145–163, 2005.

[151] P. Turaga, A. Veeraraghavan, and R. Chellappa. Statistical analysis on stiefel and grassmann manifolds with applications in computer vision. In *CVPR*, 2008.

[152] E. Ueda, Y. Matsumoto, M. Imai, and T. Ogasawara. Hand-Pose Estimation for Vision-based Human Interfaces. *IEEE Transactions on Industrial Electronics*, 50(4), 2003.

[153] R. Urtasun and T. Darrell. Sparse Probabilistic Regression for Activity-independent Human Pose Inference. *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[154] R. Varkonyi-Koczy and B. Tusor. Human-Computer Interaction for Smart Environment Applications Using Fuzzy Hand Posture and Gesture Models. *IEEE Transactions on Instrumentation and Measurement*, 60(5):1505–1514, 2011.

[155] M. Vincze, M. Zillich, W. Ponweiser, V. Hlavac, J. Matas, S. Obdrzalek, H. Buxton, J. Howell, K. Sage, A. Argyros, C. Eberst, and G. Umgeher. Integrated vision system for the semantic interpretation of activities where a person handles objects. *Computer Vision and Image Understanding*, 113(6):682–692, 2009.

[156] S.N. Vitaladevuni, V. Kellokumpu, and L.S. Davis. Action recognition using ballistic dynamics. In *CVPR*, 2008.

[157] M. Vlachos, G. Kollios, and D. Gunopulos. Discovering Similar Multidimensional Trajectories. *IEEE ICDE*, 2002.

[158] A. Waibel, R. Stiefelhagen, R. Carlson, J. Casas, J. Kleindienst, L. Lamel, O. Lanz, D. Mostefa, M. Omologo, F. Pianesi, L. Polymenakos, G. Potamianos, J. Soldatos, G. Sutschet, and J. Terken. Computers in the human interaction loop. In *Handbook of Ambient Intelligence and Smart Environments, Springer*, 2010.

[159] J. Wang, P. Liu, M. She, and H. Liu. Human Action Categorization Using Conditional Random Field. *IEEE Symposium Series on Computational Intelligence, Paris*, 2011.

[160] D. Weinland, M. Özuysal, and P. Fua. Making action recognition robust to occlusions and viewpoint changes. In *ECCV*, 2010.

[161] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *CVIU*, 104(2):249–257, 2006.

[162] D. Weinland, R. Ronfard, and E. Boyer. Action recognition from arbitrary views using 3d exemplars. In *ICCV*, 2007.

[163] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *INRIA Report*, RR-7212:54–111, 2010.

[164] N. Werghi. Segmentation and modeling of full human body shape from 3-d scan data: A survey. *TSMC-C*, 37(6):1122–1136, 2007.

[165] M. Wilder, M. Jones, and M.C. Mozer. Sequential effects reflect parallel learning of multiple environmental regularities. In Y. Bengio, D. Schuurmans, J. Lafferty, C.K.I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22, pages 2053–2061. NIPS Foundation, La Jolla, CA, 2010.

[166] P. Yan, S.M. Khan, and M. Shah. Learning 4d action feature models for arbitrary view action recognition. In *CVPR*, 2008.

[167] E. Yu and J. K. Aggarwal. Human Action Recognition with Extremities as Semantic Posture Representation. *IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

[168] Z. Zeng, M. Pantic, G.I. Roisman, and T.S. Huang. A survey of affect recognition methods: audio, visual, and spontaneous expressions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(1), 2009.

[169] J. Ziegler, K. Nickel, and R. Stiefelhagen. Tracking of the Articulated Upper Body on Multi-View Stereo Image Sequences. In *CVPR*, 2006.