UNIVERSITY OF CALIFORNIA, SAN DIEGO

A General Probabilistic Framework for Volumetric Articulated Body Pose Estimation and Driver Gesture, Activity and Intent Analysis for Human-Centric Driver Assistance

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

in

Electrical Engineering
(Signal and Image Processing)

by

Shinko Yuanhsien Cheng

Committee in charge:

Professor Mohan M. Trivedi, Chair
Professor Serge Belongie
Professor Pamela Cosman
Professor Jim Hollan
Professor Kenneth Kreutz-Delgado

2007

The dissertation of Shinko Yuanhsien Cheng is approved, and it is acceptable in quality and form for publication on microfilm:

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2007

In the memory of my grandmother.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

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

First and foremost, I would like to thank my advisor, Professor Mohan Trivedi for his unequaled kindness, unwavering encouragement, support, and enthusiasm ever since the start of my graduate career. He has provided me with much needed confidence to enable me to complete my studies. I would also like to thank the members of my committee: Professors Serge Belongie, Pamela Cosman, James Hollan, and Kenneth Kreutz-Delgado, for their advice and constructive criticism.

I would like to thank my parents, Eugene and Ammy Cheng, who have given me nothing but unconditional love and support in all the endeavors that have led to the completion of this work. I would like to thank my brother and sister, Howard and Carkay, for being such great role-models. Without them, certainly none of this would have been possible.

I would like to acknowledge my colleagues and friends from the Computer Vision and Robotics Research Laboratory for their assistance in conducting experiments and holding deep discussions relevant to this thesis, and their overall scholarship and friendship over the past several years. Specifically, I would like to thank Dr. Stephen Krotosky for the many intriguing discussions, memorable collaborations and camaraderie, Dr. Junwen Wu for the many intriguing discussions into Machine Learning, Dr. Sangho Park and Dr. Tarak Gandhi for their insightful feedback and expert advice in human activity analysis and object tracking, Dr. Kohsia Huang for showing me the ropes early on, Dr. Ivana Mikic for her inspiration and advice in articulated body pose estimation and graduate life, Mr. Erik Murphy-Chutorian, Mr. Anup Doshi, Mr. Brendan Morris, Mr. Shankar Shivappa, Mr. Ofer Achler, Mr. Giovanni Gualdi, and Mr. Dashan Gao for their help in the numerous bouts of data collection as well as many helpful technical discussions.

I would also like to thank Dr. Berkhard Huhnke Dr. Arne Stoschek, Dr. Klaus Schaff, and Mr. Jaime Camhi of Volkswagen of America, Electronics Research Laboratory for their valuable inputs. As the result of the sponsorship of VW-ERL and the UC Discovery program, I was able to freely pursue this area of research.

I owe a special thanks to David Chi for all his advice, and whose friendship I had the honor to hold throughout this nearly decade-long journey since our undergraduate years at UCSD.

I would also like to express my gratitude to someone who has helped me in less tangible but far-reaching ways. My teacher and mentor Mr. Arthur Fortgang who started me off in the field of physics and showed me the beauty of science, instilled in me the courage to take the initiative, and the courage to question, ideas which has allowed me to persevere in my studies.

Finally, I would like to thank Duangmanee Putthividhya whose love and encouragement – and technical insight – have made this pursuit so much more enjoyable and worthwhile.

La Jolla *Shinko Cheng*

Wed 7th Nov, 2007

The text of Chapter II, in part, is a reprint of the material as it appears in: Shinko Y. Cheng and Mohan M. Trivedi, "Articulated Human Body Pose Inference from Voxel Data Using a Kinematically Constrained Gaussian Mixture Model," in Proceedings and best paper award winner of the Workshop on Evaluation of Articulated Human Motion and Pose Estimation in conjunction with IEEE CVPR, 2007, and Shinko Y. Cheng, Mohan M. Trivedi, "Multimodal Voxelization and Kinematically Constrained Gaussian Mixture Model for Full Hand Pose Estimation: An Integrated Systems Approach," in Proceedings of IEEE International Conference on Computer Vision Systems, Jan. 2006, pages 34-42. I was the primary researcher of the cited materials and the co-author listed in these publications directed and supervised the research which forms the basis of this chapter.

The text of Chapter III, in part, is a reprint of the material as it appears in: Shinko Y. Cheng, Mohan M. Trivedi, "Turn-Intent Analysis Using Body Pose for Intelligent Driver Assistance", IEEE Pervasive Computing, vol. 5, number 4, pages 28-37, Oct-Dec 2006. I was the primary researcher of the cited material and the co-author listed in this publication directed and supervised the research which forms the basis of this chapter.

The text of Chapter IV, in part, is a reprint of the material as it appears in: Shinko Y. Cheng, Sangho Park, Mohan M. Trivedi, "Multi-spectral and Multi-perspective Video Arrays for Driver Body Tracking and Activity Analysis," Computer Vision and Image Understanding: Special Issue on Advances in Vision Algorithms and Systems Beyond the Visible Spectrum, vol. 106, number 2–3, pages 245-257, May-Jun. 2007. Sangho Park and I were the primary researchers of the cited material, and Professor Trivedi directed and supervised the research which forms the basis of this chapter.

The text of Appendix A, in part, is a reprint of the material as it appears in: Mohan M. Trivedi, Shinko Y. Cheng, "Holistic Sensing and Active Displays for Intelligent Driver Support Systems" IEEE Computer Magazine: Special Issue on Human-Centered Computing, 40(5):60-68, May 2007. I was the primary researcher of the Dynamic Active Display experiments, and the co-author listed in this publication was the primary researcher for the remaining parts, as well as directed and supervised the research which forms the basis of this chapter.

VITA

2001	Bachelor of Science, Electrical Engineering, University of California, San Diego
2002–2007	Graduate Student Researcher, University of California at San Diego
2003	Master of Science, Electrical Engineering, University of California, San Diego
2007	Doctor of Philosophy, Electrical Engineering, University of California at San Diego

PUBLICATIONS

S. Y. Cheng, S. Park, M. M. Trivedi, "Multi-spectral and Multi-perspective Video Arrays for Driver Body Tracking and Activity Analysis," *Computer Vision and Image Understanding: Special Issue on Advances in Vision Algorithms and Systems Beyond the Visible Spectrum,* 106(2-3):245-257, May-Jun. 2007, doi: 10.1016/j.cviu.2006.08.010

S. Y. Cheng, A. Doshi, M. M. Trivedi, "Active Heads-up Display based Speed Compliance Aid for Driver Assistance: A Novel Interface and Comparative Experimental Studies," In IEEE Proc. on *Intelligent Vehicles Symposium,* 2007.

S. Y. Cheng, M. M. Trivedi, "Articulated Human Body Pose Inference from Voxel Data Using a Kinematically Constrained Gaussian Mixture Model," In *CVPR EHuM2: 2nd Workshop on Evaluation of Articulated Human Motion and Pose Estimation,* 2007. (Winner of Best Paper Award)

S. Y. Cheng, M. M. Trivedi, "Lane Tracking with Omnidirectional Cameras: Algorithms and Evaluation," *EURASIP Journal on Embedded Systems,* Volume 2007 (2007), Article ID 46972, 8 pages doi:10.1155/2007/46972.

M. M. Trivedi, S. Y. Cheng, "Holistic Sensing and Active Displays for Intelligent Driver Support Systems," *IEEE Computer Magazine: Special Issue on Human-Centered Computing,* 40(5):60-68, May 2007.

S. Y. Cheng, M. M. Trivedi, "Turn-Intent Analysis Using Body Pose for Intelligent Driver Assistance" *IEEE Pervasive Computing,* 5(4):28-37, Oct-Dec 2006.

S. Y. Cheng, M. M. Trivedi, "Multimodal Voxelization and Kinematically Constrained Gaussian Mixture Model for Full Hand Pose Estimation: An Integrated Systems Approach" In Proc. of *IEEE International Conference on Computer Vision Systems,* Jan. 2006, pages 34-42.

S. Y. Cheng, S. Park, M. M. Trivedi, "Multi-Perspective Thermal IR and Video Arrays for 3D Body Tracking and Driver Activity Analysis" In Proc. of *IEEE International Workshop on Object Tracking and Classification in and Beyond the Visible Spectrum in Conjunction with IEEE International Conference on Computer Vision and Pattern Recognition,* Jun. 2005.

S. J. Krotosky, S. Y. Cheng, M. M. Trivedi, "Real-Time Stereo-Based Head Detection using Size, Shape and Disparity Constraints" In Proc. of *IEEE International Symposium on Intelligent Vehicles,* Jun. 2005.

M. M. Trivedi, S. Cheng, E. Childers, S. J. 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), Nov. 2004.

S. Y. Cheng, M. M. Trivedi, "Human Posture Estimation Using Voxel Data for 'Smart' Airbag Systems: Issues and Framework" In Proc. of *IEEE International Symposium on Intelligent Vehicles,* May 2004, Pages:84-89.

S. J. Krotosky, S. Y. Cheng, M. M. Trivedi, "Face Detection and Head Tracking using Stereo and Thermal Infrared Cameras for 'Smart' Airbags: A Comparative Analysis," In Proc. of *IEEE Conference on Intelligent Transportation Systems,* Mar. 2004.

ABSTRACT OF THE DISSERTATION

A General Probabilistic Framework for Volumetric Articulated Body Pose Estimation and Driver Gesture, Activity and Intent Analysis for Human-Centric Driver Assistance

by

Shinko Yuanhsien Cheng
Doctor of Philosophy in Electrical Engineering (Signal and Image Processing)
University of California, San Diego, 2007
Professor Mohan M. Trivedi, Chair

In this thesis, we investigate ways of enabling intelligent systems to recognize human desires and wants, and we devise systems that automatically recover human pose and gesture information. We place special emphasis on applications for improving the safety and comfort of vehicles.

We present a novel method for learning and tracking the pose of an articulated body by observing only its volumetric reconstruction from images. The model is called the kinematically constrained Gaussian mixture model (kc-gmm). Pairs of components connected at a joint are encouraged to assume a particular spatial configuration, forming joints with 1, 2 or 3 degrees-of-freedom (DOF). Pose learning is based on the EM algorithm, and is the first to be evaluated using a common human image data-set with optical motion capture ground-truth. The algorithm achieved estimates with mean joint position error of 15.9 cm, or 8% of the total length of the body. On synthesized hand data, the error was 0.5 cm, or 1.5% of the total length.

Next, we present results on the characterization and recognition of driver intent using driver gestural cues. The concepts apply towards the study of other driving maneuvers. The data-driven pattern classification approach makes use of vehicle dynamics information and driver head and hand pose information via an optical motion capture system. We present results comparing different combinations of input cues. We proposed a novel visualization of results to analyze the classifiers: ROC Area vs. Decision Time and Statistical Response Over Time plots.

Driver-intent recognition algorithm above assumes the use of body part position information. We present an in-vehicle system for detecting and tracking the position of the left and right hands in long-wavelength infrared imagery. The results were effective in tracking hands over 90 minutes of driving. Combined with steering information, 5 hand activities over the steering wheel could also be determined.

Finally, we present an in-vehicle system for determining which occupant is accessing the vehicle infotainment controls for modulating information flow from the vehicle's information display. The average correct classification rate of 97.8% was achieved over 60 minutes of 30fps video under a variety of moving vehicle operating conditions.

I

Introduction

This introductory chapter presents the motivation behind our work, the formulation of the research problems for which we propose solutions, and our contributions. The chapter also includes an overview of the thesis.

I.1 Motivation

Human gestures are the meaning conveyed through the pose and movement of the human body. Gesture is said to be as rich and varied as spoken language itself, manifesting from the physical human body as ideas, interests, feelings and intentions [1]. From the perspective of human computer interface (HCI) research, there is much to be gained from a thorough understanding of gesture and pose to improve the efficiency of the flow of information between the ever ubiquitous computer and its human operator. Such an understanding facilitates the communication between them by allowing for more natural methods of generating commands [2]. A special quality of some gestures is that they can be involuntarily "uttered" in the course of human thought. Those thoughts could conceivably be inferred from observing the pose of the human body. Together, inferred thought and explicit commands give an artificial system the ability to efficiently collect and reliably guess the needs and desires of the human operator.

A particular set of human gestures that have gained much attention from researchers of intelligent vehicles are driving gestures. In HCI terminology, we refer to both the communicative and manipulative gestures of driving [2]. An understanding of these human gestures has significant implications for the safety and comfort of an activity that is ever present in our lives. It is expected that 2.3 million vehicle crashes and over 43,000 fatalities will occur on U.S. roads in 2007 [3]. Driver inattention was determined to be the contributing factor in an estimated 32% of all collisions between 2002 and 2003 [4]. Inattention is characterized by tasks secondary to driving, consisting of wireless device access, activity

involving passengers, interior distractions, personal hygiene and dining as the primary types [5]. Driver gesture recognition would enable the vehicle's computer to automatically determine whether the driver is being inattentive in nearly 700,000 of the collision situations expected to occur. Upon determining that a driver was inattentive, the computer can be tasked with co-pilot-like responsibilities to assist and warn the driver upon detecting dangerous situations, thus making the vehicle safer to drive. To realize these potential benefits, the primary task is therefore to understand the driver and the driver's gestures.

For an example of how gesture recognition may help, imagine the scenario in which a motorist tries to turn right at an intersection and a bicyclist is in his blind spot. If the vehicle could detect the bicyclist in its path, it could alert the driver and potentially avoid the collision. However, if the driver has already seen the bicyclist, an alert might hinder rather than complement safe driving by unnecessarily increasing the driver's workload. Likewise, if the vehicle knew that the driver failed to notice the bicyclist, it could more confidently alert the driver of the danger, or even take over the driving if it is confident enough. For an intelligent driver-assistance system to be effective, it must be able to continuously monitor not just the surrounding environment and vehicle state, but also monitor the driver's gestures. If a dangerous situation occurs that requires intervention, the vehicle can recognize it and alert the driver more accurately.

The driver pose estimation aspect of gesture recognition is by itself not limited to applications in intelligent vehicles. Body pose estimation, or human motion capture as it is more widely known, has a number of applications, including video surveillance systems, video analytics (store visitor analysis), intelligent rooms, human computer interfaces, ergonomics studies, gait pathology studies, sports tuning, robot control, and 3-D animation. The pose estimation problem involves estimating the parameters of the human body model (such as joint angles) from sensed data. The sensors may be laser-range scanners, time-of-flight sensors, long-wavelength infrared imagers or broad-spectrum visible wavelength imagers (video cameras). The use of multiple sensors has been explored as well, capturing different perspectives of the same scene. Each of these sensors provides 2-D or 3-D images of the scene, which a model would then relate to the pose of the subject.

The processing flow of a gesture recognition system generally begins with sensing the subject and extracting pertinent sensor data primitives or cues for estimating the pose, which are finally used in determining the subject's gestures. Tab. I.1 illustrates this basic process of recognizing human gestures with the types of results from the intermediate processing steps.

Systems that provide the ability of recognizing driving gestures in actual vehicles require specific functionality and features that are not provided by similar systems proposed to-date which do not have this goal in mind. Also, it is unknown to what extent gestures will contribute to improvement in intelligent vehicles. In this thesis, we focus on the design of pose and gesture recognition techniques with emphasis on intelligent vehicles.

Table I.1: Abstraction layers of information contained in the physical human body.

Raw Sensed Data	Sensor Features	Pose	Gesture
Long-wavelength Infrared Images	Edges	Skeletal Structure	Posture
	Appearance	Limb Dimensions	Person Identity
Monochrome color cameras	Invariant Moments	Joint Location	Facial, Hand, Body Gesture
	Depth Maps	Joint Range -of-motion	Facial Expression
Time-of-flight Imagers (LIDAR)	Landmarks descriptors	Face Location	Gait
Laser-range scanner	Region descriptors	Body part Pose (e.g. Head)	Activity
	Volume		Intent

I.2 Problem Statement and Challenges

The first challenge in driver gesture recognition is to identify and characterize the kinds of driver gestures that enable a vehicle to determine the attention or comfort level of the driver. For this thesis, we focus on developing the techniques for recognizing the driver's intention to make an intersection turn, and determine the occupant whose hand is hovering over the controls in the aisle console. Both kinds of gestures are important for driving safety: intersection-turn intent can generate more accurate warnings in collision situations, and user determination helps drivers focus on the driving by preventing their access to the increasingly popular vehicle infotainment system, while allowing the passenger to access it.

In order to recognize these driver gestures, and gestures in general, the system requires some description of the driver's body pose. Depending on the needs of the application, the description can explicitly describe the skeletal configuration of the driver's body, i.e. joint position, joint type, body part orientation, or be simplified to just describe position or appearance. The type of description that is both adequate and optimal needs to be understood.

All of the system components must adhere to special properties of the vehicular environment which introduces very specific requirements for these systems. For vision sensors, external lighting conditions will influence the behavior of the captured images of the scene and therefore require a processing algorithm that is invariant or robust to such illumination changes.

To conclude, we focus on developing gesture recognition systems that also fulfill vehicle-specific requirements related to intersection-turn intent and user determination. In the process, we will also investigate improvements to the general articulated body pose estimation problem and modifications for its use in vehicles.

I.3 Contributions

This thesis contributes to the area of pose estimation, gesture recognition, and human-centric driver assistance systems for enhancing driving safety and comfort. The main questions we address are:

- What precisely are the driver gestures we can recognize?
- How can driver intersection-turn intent be reliably recognized?
- To what extent does body pose information improve the recognition performance of driver intent, if any?
- How can driver body-part position be tracked in the vehicle without the use of encumbering markers or wires?
- How can the user of the vehicle controls be determined by observing images of the hand accessing the controls?

The detailed contributions of this thesis include:

1. A modeling framework and learning procedure that relates volumetric reconstructions of articulated bodies with the pose of the body using the kinematically constrained Gaussian mixture model (kc-gmm). This approach was evaluated using motion-capture generated ground-truth as part of a common data-set. The results were competitive with other approaches.

2. A system for recognizing the driver intent to perform a "slow" intersection turn maneuver using the kernel Relevance Vector Machine classifier and several time-series cues including vehicle dynamics and driver head pose and hand position. A study of intersection-turn trends in vehicle and driver data and the pertinence of various cues are also presented.

3. A hand position tracker from thermal infrared images using cascade of boosted classifiers working with haar-like features, and multiple hand tracking with a probabilistic data association filter used for steering wheel grasp analysis and driver activity recognition.

4. A vision-based user determination system which determines which of the two front-row seat occupants, if anyone at all, is accessing the controls in the center console.

I.4 Thesis Outline

The rest of this thesis is organized as follows:

Chapter II presents a general articulated body pose estimation algorithm using the kinematically constrained Gaussian mixture model to represent voxel images of bodies and the EM algorithm as the estimator.

Chapter III presents the proposed driver gesture recognition system, namely a driver intersection-turn intent inference system.

Chapter IV presents a vision-based approach for estimating the position of the driver's hands using thermal imagery in the vehicle.

Chapter V presents vision-based approaches to estimating the 2-D image and 3-D spatial position of the driver's hands in the vehicle.

Chapter VI summarizes the work and presents final remarks. This chapter also includes future research directions.

II

Articulated Body Pose Estimation

In our first contribution towards pose and gesture recognition, we present our study on estimating the pose of a generic articulated body.

II.1 Introduction

We present a novel method for learning and tracking the pose of an articulated body by observing only its volumetric reconstruction or "voxel image". These voxel images are the kind that can be derived from a set of images of the subject captured from various perspectives and calculated via shape-from-silhouette or other techniques [6]. The primary challenge is devising a method to efficiently and robustly extract joint parameters that describe the pose of the articulated body from unlabeled 3D voxel image sequences.

This work stems from the desire to develop accurate tether-less, vision-based articulated body pose estimation systems. These bodies may be the human body, hand, or articulated structures. Such a system has several foreseeable applications, including marker-less motion capture for human-computer interfaces, physiotherapy, 3D animation, ergonomics studies, robot control and surveillance. One of the major difficulties in recovering pose from images of an articulated body is the high number of degrees-of-freedom (DOF) in movement that needs to be recovered. Any rigid object requires 6 DOF to fully describe its pose. Each additional rigid object connected to it adds at least 1 DOF. A human body contains no less than 10 large rigid body parts, equating to more than 20 DOFs. The difficulty is compounded by the problem of self-occlusion, where body parts occlude each other depending on the configuration. Other challenges involve dealing with varying illumination which affects appearance, varying subject attire or body type, required camera configuration, and required computation time.

One framework is to first extract the volume representation of the articulated body from image

data. Images of the subject may be segmented, generating a silhouette image of the subject. The resulting silhouettes can then be back-projected into a series of camera rays through the silhouette back onto the scene. The intersection of these rays from many images from different perspectives of the subject would constitute the visual-hull of the subject, an upper-bound to the actual volume of the articulated body [7,8]. Even tighter bounds can be achieved if the colors or texture of the surfaces are taken into account [7,9]. From here, the volumes are used to extract the pose of the body. This isolates the problem of finding the pose of the articulated body from voxel data from the problem of computing the volume reconstruction. The proposed pose learning method assumes the use of only the voxel data acquired using shape-from-silhouette or other volumetric reconstruction technique.

We propose a probabilistic technique that utilizes a multi-component Gaussian mixture model to describe the spatial distribution of voxels in a voxel image. Each component describes a segment or rigid body, and the collection of components are kinematically constrained according to a pre-specified skeletal model. This model we refer to as a kinematically constrained Gaussian mixture model (kc-gmm). The kinematic constraints are in the form of a probability density function that gives a high probability when pairs of components connected at a common joint satisfies a particular spatial configuration, forming a 1, 2 or 3 degree-of-freedom (DOF) joint. This is done by incorporating a constraint function as a prior on the component means, which represent the components' location in $\mathbb{R}^3$, and the covariance matrix, which represents the orientation of the component. Component rotation is parameterized in terms of Euler angles. All parameters are learned using the EM algorithm.

The pose learning algorithm is evaluated using synthesized hand data, and the HumanEvaII data-set for facilitating comparison among different algorithms. Both data-sets contain ground-truth information for accuracy measurements. For the case of the hand, we illustrate hand pose learning using a 16 component, 27 DOF mixture model. For the human body in the HumanEvaII data-set, we illustrate human body pose learning using a 11 segment, 19 DOF mixture model. The results show that utilizing volume data and aided by the degrees-of-freedom constraints only, this approach attains accuracies of joint location estimates within 0.5cm mean-absolute-error from ground-truth with the hand data set and 17cm from subjects S2 and S4 from the HumanEvaII data-set. (S2 and S4 are subjects comprises of all the subjects in HumanEvaII data-set. The HumanEvaI data-set consists of more subjects, including S2 and S4, using a different video collection hardware.)

The statistical model lends itself easily to the estimation of the two attributes of the articulated body simultaneously: body structure and body pose, as both are parameters within this model. The model parameters describe 1) the dimensions of each component in height, width and depth and 2) the location and orientation of each segment or equivalently, the joint angles and component position. For this chapter, the focus is on recovering body pose. However, we hope these results will serve as an indicator of the promise that statistical clustering techniques of volume data can be used to resolve more than body pose.

In the following sections, we lay the foundation for the proposed algorithm, describing the previous research in this area in sec. II.2. The model and learning procedure are described in sec. II.3 and II.4. To demonstrate the effectiveness of the proposed technique, the pose learning algorithm is evaluated on two sets of data with ground-truth: synthetic hand data and HumanEvaII data-set. The results of these tests are described in sec. II.5. Finally, the chapter concludes with discussion in sec. II.6.

II.2 Related Work

There has been a tremendous amount of work in image-based recovery of articulated body pose. Several surveys of such techniques can be found in [2, 10–12]. Numerous ways have been devised to represent pose as a function of volumetric data. Each consists of a model and a fitting procedure to fit the model to the data. One of the earlier works is by Cheung *et al.* [13], where a simple k-means like algorithm is used to estimate the torso and 5 major appendages of the body (head, arms and legs). Largely to demonstrate the real-time volume reconstruction technique, no actual kinematic model was assumed.

Mikic *et al.* [14] devised a method of tracking articulated human body hierarchically, starting by detecting the head, then fitting a torso attached to the head. Then the remaining voxels are segmented to locate the upper and lower legs and arms. The strength of the Mikíc approach is an initialization procedure to the tracking process, which the proposed kc-gmm method in its current state does not have. Mikic's approach however lacks generality to extend tracking articulated objects of an arbitrary skeletal structure. Furthermore, Mikic's approach can be described as top-down in nature and the proposed approach is bottom-up. The result is that limbs at the end of the hierarchy contribute to the estimate of the whole body as much as other parts that are higher up in the hierarchy, ultimately converging at a compromise among all components.

The research most closely related to this chapter is that of the constrained mixture model work by Hunter *et al.*, [15]. They too utilize the concept of constraining the configuration of Gaussian components in a mixture model but in 2-D silhouette images. We extended the Hunter model to describe volumes in [16]. In both these works, the model parameters were learned utilizing Expectation-Constrained-Maximization. This estimation procedure involves injecting a constraining step following the E- and M-steps to project the parameter estimates onto the kinematically feasible manifold. It is conceivable that the M-step may conflict with the so-called constraining C-step, causing instability in the optimization process. Our primary contribution in this chapter is incorporating these kinematic constraints (confining pairs of components to have non-zero rotation along only the specified degrees-of-freedom) into the probability model in the form of a parameter constraining prior probability. This allows us to remove the C-step completely, stay within the EM algorithm framework, and enjoy all the proven convergence properties as a result.

Two other noteworthy approaches in the volume-based pose estimation area are by Ueda *et al.* [17] and Ogawara *et al.* [18]. Their techniques are based on the iterative-closest-point (ICP) algorithm. The differences between our approach and theirs in this case are subtle. Each algorithm arrives at the pose estimate result with roughly the same accuracy. Their approach utilizes the actual volume reconstruction itself as part of their model in which they position the joints and divide the volume into segments. In contrast, our approach requires knowing only the dimensions of the individual segments, and does not require a representative volume reconstruction for the algorithm to operate in subsequent frames. Thereby, in cases where the volume data of the articulated object we wish to track is unknown before hand, e.g. partially visible driver in a car, tracking can still take place with our approach. The kc-gmm approach does not currently address the issue of adaptive dimension adjustment according to the volume data; however, the primary motivation of this approach is that body structure can also be recovered using the same paradigm of probabilistic clustering.

The most important motivation for using a probabilistic mixture model to describe volumetric reconstructions of bodies is that it can conceivably allow easier estimation of body structure, which is thus-far the most elusive articulated body attribute to learn from image data. Humans can discern one rigid body part from another quite easily by examining a sequence of voxel images of a moving articulated body and determining which voxels move together with respect to others. The moving volume cue alone should be adequate to determine joint locations, joint type, and body part dimensions for many applications. Some deterministic but automatic ways have been presented illustrating this concept of grouping rigid parts from 3-D visual-hull [19,20], 3-D color surface points [21], and 2-D image point data [22]. There is promise that a mixture model approach will serve as a basis to learn body structure. This chapter solves the first problem of estimating kinematically feasible pose using this probabilistic model, leaving structure learning for future work.

II.3 Kinematically Constrained Gaussian Mixture Model

The kinematically constrained Gaussian mixture model consists of the usual mixture of Gaussians model [23] with a prior probability on the constraints which in turn influences the mixture parameters. Fig. II.1 shows a graphical representation of the model. If we let y_n be distributed by a mixture of K Gaussians representing K rigid body parts, z_n be the hidden membership variable, and Θ be the embodiment of the kinematic constraints and all means and covariance matrices of every Gaussian density, the

Table II.1: Relevant works on human body pose estimation using silhouettes and voxel reconstructions.

	Image Feature	Model	Pose Estimation Method	Object Type	Evaluation	Comments
Hunter *et al.*('99) [24]	Monocular Silhouette	Gaussian mixture model with feasible kinematic configuration space projection	ECM	Coarse Body	Subjective	Several assumptions due to monocular input.
Rosales *et al.*('00) [23]	Monocular Silhouette Hu moments	Specialized Mappings	Pose classifier Silhouette Matching as Criteria	Hand	Ground-truth from glove device	Operates best on hands consistent/similar with training data.
Delamarre *et al.*('01) [23]	Multiple Silhouettes	3-D shape primitive model	ICP & Natural/Spring forces	Body	Subjective	Limited Evaluation
Ogawara ('03) [23]	voxel image	Surface and skeletal hand model.	ICP, M-estimator (3DTM)	Hand	Subjective & Ground-truth from Synthesis	Addresses holes in voxel image.
Ueda ('01, '03) [23]	voxel image	surface and skeletal hand model.	Torque forces induced by exposed surface model points.	Hand	Subjective	-
Mikic ('02-'03) [23]	voxel image	ellipsoidal components skeletal structure described with twists	Kalman Filter Bayesian Network	Body	Subjective	-
Cheung ('03) [21]	CSP	CSP cloud	CSP alignment	body	Subjective	-
This Approach [16, 25]	voxel image	kc-gmm	EM algorithm	hand & body	Subjective & ground-truth from synthesis and mocap	First volumetric approach evaluated with mocap ground-truth.

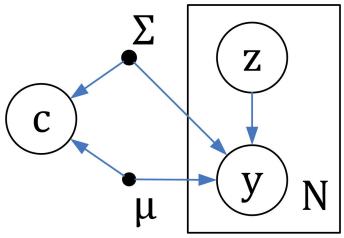


Figure II.1: Graphical representation of the kinematically constrained mixture model.

density function of $Y = \{\mathbf{y}_n\}_{n=1}^N$ has the form

$$\begin{aligned} P(Y, c|\Theta) &= P(c|\Theta) \prod_n P(\mathbf{y}_n|c, \Theta) \\ &= P(c|\Theta) \prod_n \left[\sum_{z_n} P(\mathbf{y}_n|z_n, \Theta) P(z_n) \right] \\ &= P(c|\Theta) \prod_n \left[\sum_{z_n} \mathcal{N}(\mathbf{y}_n|\mu_{z_n}, \Sigma_{z_n}) \pi_{z_n} \right] \end{aligned} \tag{II.1}$$

The expression in square brackets is the familiar mixture model. We introduce a zero-mean normally distributed random variable c which constrains components pairwise. There are altogether three forms of these constraints: spherical (3-DOF) constraint, hardy-spicer (2-DOF) constraint, and revolute (1-DOF) constraint.

Any two components connected by a joint is constrained using the spherical constraint given by

$$c_s(\Theta) = \mu_i + \mathbf{R}_{0i}\mathbf{a}_{ij} - (\mu_j + \mathbf{R}_{0j}\mathbf{a}_{ji}) \tag{II.2}$$

where $\mu_i, \mu_j \in \mathbb{R}^3$ are the means of components i and j, $\mathbf{R}_{0i}, \mathbf{R}_{0j} \in \mathrm{SO}(3)$ are the rotation of the components relative to the world coordinate frame, and $\mathbf{a}_{ij}, \mathbf{a}_{ji}$ point to the joint location from the component centers in component coordinate frame. This constraint represents a path from the origin, to the center of one component (μ_i), to the joint shared between the two components, to the center of the other component (μ_j), and back to the origin. $C_s(\Theta)$ equals zero if the two components meet at the joint. Likewise, the other two constraints operate in the same manner; when the constraint given the component means and orientations equals zero, the DOF constraint is satisfied.

The hardy-spicer constraint is given by

$$c_h(\Theta) = \mathbf{R}_{0i}\mathbf{q}_{ij} \cdot \mathbf{R}_{0j}\mathbf{q}_{ji} \tag{II.3}$$

where $\mathbf{q}_{ij}, \hat{\mathbf{q}}_{ji}$ are the rotational axes of each component in either component coordinate frame. In this case, they each equal one of the two rotational axes. For example, if $\mathbf{q}_{ij} = (1, 0, 0)$ and $\mathbf{q}_{ji} = (0, 1, 0)$, the joint between the two components i and j is a 2-DOF joint that can rotate along the x- and y-axes with respect to either component coordinate frame.

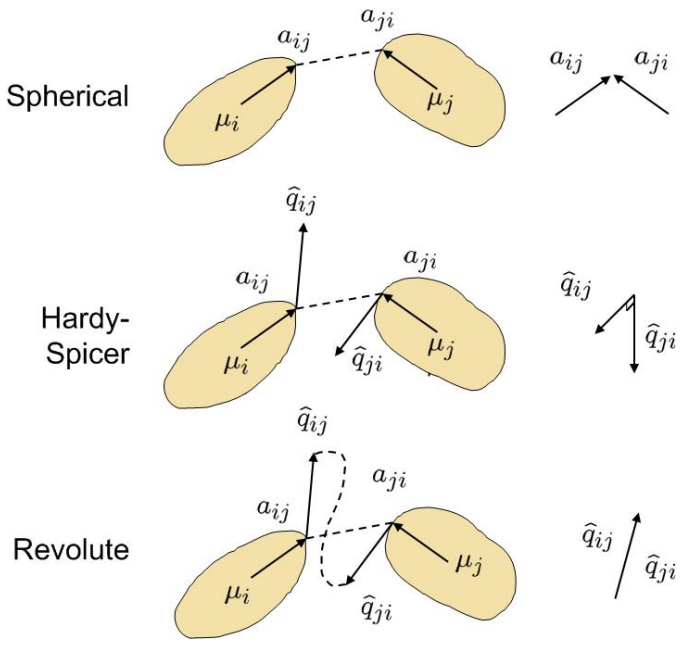


Figure II.2: Illustration of the joint constraints. The right column illustrates the configuration of joint location vectors $\mathbf{a}_{ij}, \mathbf{a}_{ji}$ and rotational axes $\mathbf{q}_{ij}, \mathbf{q}_{ji}$ when the constraints are satisfied. The length of the dotted lines are reduced to zero when the constraints are satisfied

The revolute constraint is given by

$$c_r(\Theta) = \mathbf{R}_{0i}\mathbf{q}_{ij} - \mathbf{R}_{0j}\mathbf{q}_{ji} \tag{II.4}$$

Again, $\mathbf{q}_{ij}, \mathbf{q}_{ji}$ represent the rotational axes. When this constraint is satisfied, the two rotational axes align resulting in a rotation along only the single DOF. Usually, $\mathbf{q}_{ij} = \mathbf{q}_{ji}$, although this need not be the case.

Fig. II.2 illustrates the mechanics of the constraints. To describe what occurs in each constraint physically, for every joint in the articulated body, c_s pulls the pair of components together to make contact at the specified joint, then c_h and c_r orients the components, possibly even translating the components such that the relative rotation between the two components are non-zero along the 2 and 1 degrees-of-freedom, respectively.

Finally, $\mathbf{R}_{0i}$ and $\mathbf{R}_{0j}$ are extracted from mixture components by parameterizing the covariance

matrix of the Gaussian densities in the following way:

$$\begin{aligned}
\mathbf{y}_n &\sim \sum_{i=1}^{K} \mathcal{N}(\mathbf{y}_n|\mu_i, \Sigma_i)\pi_i \\
\Sigma_i &= \mathbf{R}_i \Lambda_i \mathbf{R}_i^\top \\
&= \mathbf{R}_{zi}\mathbf{R}_{yi}\mathbf{R}_{xi}\Lambda_i\mathbf{R}_{xi}^\top\mathbf{R}_{yi}^\top\mathbf{R}_{zi}^\top \\
&= e^{\hat{\mathbf{z}}\theta_{zi}} e^{\hat{\mathbf{y}}\theta_{yi}} e^{\hat{\mathbf{x}}\theta_{xi}} \Lambda_i e^{-\hat{\mathbf{x}}\theta_{xi}} e^{-\hat{\mathbf{y}}\theta_{yi}} e^{-\hat{\mathbf{z}}\theta_{zi}} \\
&= \Sigma(\theta_{xi}, \theta_{yi}, \theta_{zi})
\end{aligned} \tag{II.5}$$

The matrices $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{z}}$ are skew-symmetric matrices of axes of rotation, which are along the x-, y- and z-axes in the world coordinate frame.

II.4 Learning Pose using EM

The maximum likelihood estimate of the pose is found using the EM algorithm. The E-step remains the same as the E-step for the standard Gaussian mixture model. The M-step becomes an optimization over the log-likelihood of the mixture model summed with the log-likelihood of c, the constraint on the component position (means) and orientation (covariance matrices). A closed-form expression for the mean can be found, but the orientation of each component is estimated using gradient ascent.

Using equ. II.1, the problem of finding the ML estimate of Θ can be stated as maximizing the following log-likelihood equation:

$$\hat{\Theta}_{ML} = \arg\max_{\Theta} \left[\ln P(Y|\Theta) + \ln P(c|\Theta)\right] \tag{II.6}$$

By introducing the 1) hidden membership variable z_n, 2) its distribution function $q(z_n)$ as of yet unknown, and 3) the posterior of z_n, and then rearranging the terms, we reveal the expression equivalent to the log-likelihood which can then be more readily maximized.

$$\begin{aligned}
&\ln P(Y|\Theta) \\
&= \sum_n \ln P(\mathbf{y}_n|\Theta) \sum_{z_n} q(z_n) - KL(q\|p) + KL(q\|p) \\
&= \sum_n \sum_{z_n} \left[q(z_n) \ln \frac{P(\mathbf{y}_n, z_n|\Theta)}{q(z_n)} - q(z_n) \ln \frac{P(z_n|\mathbf{y}_n, \Theta)}{q(z_n)} \right] \\
&= \mathcal{L}(q, \Theta) + \mathrm{KL}(q\|p)
\end{aligned} \tag{II.7}$$

$\mathrm{KL}(q\|p)$ is the Kullback-Liebler divergence (relative entropy), $\sum_{z_n} q(z_n) = 1$, and $\mathcal{L}$ is the so-called lower-bound of the incomplete log-likelihood. Finally substituting equ. II.7 back into equ. II.6, we arrive

at the desired expression.

$$\begin{aligned} &\ln P(Y|\Theta) + \ln P(c|\Theta) \\ &\quad = \mathcal{L}(q,\Theta) + \mathrm{KL}(q\|p) + \ln P(c|\Theta) \\ &\quad \geq \mathcal{L}(q,\Theta) + \ln P(c|\Theta) \end{aligned} \tag{II.8}$$

Maximizing the log-likelihood lower-bound $\mathcal{L}(q,\Theta)$ is equivalent to maximizing the log-likelihood. To find the $\hat{\Theta}_{\mathrm{ML}}$, we iteratively hold the parameters Θ fixed and find the distribution function q that maximizes the equation (E-step), then hold q fixed and find Θ that maximizes the log-likelihood(M-step). Details of the EM algorithm and its derivation can be found in [26].

II.4.A E-Step: Solving for $q(z_n)$

The E-Step consists of evaluating the posterior probability of the hidden variable z_n while holding the parameters fixed.

$$\begin{aligned} q(z_n) &= p(z_n\,|\mathbf{y}_n, \Theta^{\mathrm{old}}) \\ &= \frac{\mathcal{N}(\mathbf{y}_n|\mu_{z_n}, \Sigma(\theta_{z_n}))\pi_{z_n}}{\sum_{z_n}\mathcal{N}(\mathbf{y}_n|\mu_{z_n}, \Sigma(\theta_{z_n}))\pi_{z_n}} \\ &= \alpha_{z_n,i} \end{aligned} \tag{II.9}$$

When $q(z_n)$ equals the posterior of z_n, the KL divergence $KL(q\|p)$ equals zero while maintaining the same values for the incomplete log-likelihood.

II.4.B M-Step: Solving for π_i and μ_i

The M-step consists of maximizing $\mathcal{L}(q,\Theta)$ over Θ.

$$\hat{\Theta}_{\mathrm{ML}} = \arg\max_{\Theta} \mathcal{L}(q^{\mathrm{old}}, \Theta) \tag{II.10}$$

$$= \arg\max_{\Theta} \sum_n \sum_{z_n} q^{\mathrm{old}} \ln P(\mathbf{y}_n, z_n|\Theta) + \ln P(c|\Theta) \tag{II.11}$$

The parameters Θ consist of the means of each component μ_i, the orientation of each component in Euler angles θ_i, and the class prior probability π_i for all components i.

Because $P(c|\Theta)$ does not depend on the class prior probabilities π_i, they can be found the same way as learning them for the standard mixture model by evaluating

$$\hat{\pi}_i = \frac{1}{N}\sum_i \alpha_{z_n,i} \tag{II.12}$$

where N is the number of voxels.

Only the incomplete log-likelihood and spherical constraint probability $P(c_s|\Theta)$ depend on the component mean μ_i. A single component may have one or several joints constrained by the spherical joint constraint probability. Solving for μ_i involves setting the gradient of $\mathcal{L}(q,\Theta)$ with respect to all μ_i to equal zero, and solving for μ_i for all i simultaneously using Least Squares. The gradient of the log-likelihood and spherical constraint probability is given by

$$\begin{aligned}
&\nabla_{\mu_i} \sum_n \ln P(\mathbf{y}_n|z_n = i,\Theta) \\
&\quad = \sum_n \alpha_{i,n}\Sigma(\theta_i)^{-1}(\mathbf{y}_n - \mu_i) && \text{(II.13)} \\
&\nabla_{\mu_i} P(c_s|\Theta) \\
&\quad = -\nabla_{\mu_i} c_s^\top \Sigma_{c_s}^{-1} c_s \\
&\quad = -\Sigma_{c_s}^{-1}(\mu_i + \mathbf{R}_i\mathbf{a}ij - (\mu_j + \mathbf{R}_j\mathbf{a}_{ji})) && \text{(II.14)} \\
&\nabla_{\mu_j} P(c_s|\Theta) \\
&\quad = +\nabla_{\mu_j} c_s^\top \Sigma_{c_s}^{-1} c_s \\
&\quad = +\Sigma_{c_s}^{-1}(\mu_i + \mathbf{R}_i\mathbf{a}ij - (\mu_j + \mathbf{R}_j\mathbf{a}_{ji})) && \text{(II.15)}
\end{aligned}$$

Through judicious rearrangement of terms, one can construct a system of equations for this component mean μ_i and all other component means.

$$\begin{aligned}
&\left[-\Sigma_{c_s}^{-1} \cdots \sum_n \alpha_{i,n}\Sigma(\theta_i)^{-1} + \gamma\Sigma_{c_s}^{-1} \cdots - \Sigma_{c_s}^{-1}\right] \begin{bmatrix} \vdots \\ \mu_j \\ \vdots \\ \mu_i \\ \vdots \\ \mu_k \\ \vdots \end{bmatrix} \\
&= \sum_n \alpha_{i,n}\Sigma(\theta_i)^{-1}\mathbf{y}_n \\
&\quad + \Sigma_{c_s}^{-1}(\mathbf{R}_j\mathbf{a}_{ji} - \mathbf{R}_i\mathbf{a}_{ij}) + \Sigma_{c_s}^{-1}(\mathbf{R}_i\mathbf{a}_{ik} - \mathbf{R}_k\mathbf{a}_{ik}) && \text{(II.16)}
\end{aligned}$$

The component means $\hat{\mu}_i \quad \forall i$ that maximize the log-likelihood can be found using least squares to solve the system of equations shown in equ. II.16.

II.4.C M-Step: Solving for θ_i

Finally, to solve for the orientation θ_i of each component, we need to consider the incomplete log-likelihood and all relevant constraint probabilities $P(c_s|\Theta)$, $P(c_h|\Theta)$, and $P(c_e|\Theta)$. Gradient ascent

is employed for this task. The gradient of $\mathcal{L}(q^{\text{old}}, \Theta)$ with respect to θ_i is used to iteratively step toward the solution until convergence using the update equation

$$\theta_i^{[n+1]} = \theta_i^{[n]} + \alpha_n \nabla_{\theta_i} \mathcal{L} \tag{II.17}$$

The gradient of $\mathcal{L}(q^{\text{old}}, \Theta)$ with respect to θ_i is given by

$$\begin{aligned} \nabla_{\theta_i} \mathcal{L} &= \nabla_{\theta_i} \ln P(\mathbf{y}_n|\Theta) \\ &+ \nabla_{\theta_i} \left[\ln P(c_s|\Theta) + \ln P(c_h|\Theta) + \ln P(c_e|\Theta)\right] \end{aligned} \tag{II.18}$$

All constraints are included in equ. II.18, but each joint will utilize at most two of the above constraints. The gradient of the other constraints will equate to zero in those cases. All constraint probabilities are shown here to illustrate the positioning of the gradients.

The gradient of the incomplete likelihood is given by

$$\nabla_{\theta_i} \ln P(\mathbf{y}_n|\Theta) = -\sum_n \frac{\alpha_{i,n}}{2} \nabla_{\theta_i} m_i(\theta_i) \tag{II.19}$$

where

$$\begin{aligned} &\nabla_{\theta_i} m_i(\theta_i) = \\ &\begin{bmatrix} 2(\mathbf{y}_n - \mu_i)^\top e^{\hat{\mathbf{z}}\theta_z} e^{\hat{\mathbf{y}}\theta_y} \hat{\mathbf{x}} e^{\hat{\mathbf{x}}\theta_x} \Lambda^{-1} \mathbf{R}^\top (\mathbf{y}_n - \mu_i) \\ 2(\mathbf{y}_n - \mu_i)^\top e^{\hat{\mathbf{z}}\theta_z} \hat{\mathbf{y}} e^{\hat{\mathbf{y}}\theta_y} e^{\hat{\mathbf{x}}\theta_x} \Lambda^{-1} \mathbf{R}^\top (\mathbf{y}_n - \mu_i) \\ 2(\mathbf{y}_n - \mu_i)^\top \hat{\mathbf{z}} e^{\hat{\mathbf{z}}\theta_z} e^{\hat{\mathbf{y}}\theta_y} e^{\hat{\mathbf{x}}\theta_x} \Lambda^{-1} \mathbf{R}^\top (\mathbf{y}_n - \mu_i) \end{bmatrix} \end{aligned} \tag{II.20}$$

The gradient for the constraint probabilities all follow the form

$$\begin{aligned} \nabla_{\theta_i} \ln P(c|\Theta) &= \nabla_{\theta_i} \ln \mathcal{N}(c|\mathbf{0}, \Sigma) \\ &= \nabla_{\theta_i} \left[-\frac{1}{2} c^\top \Sigma^{-1} c\right] \\ &= -(\nabla_{\theta_i} c)\Sigma^{-1} c \end{aligned} \tag{II.21}$$

Depending on the ordering of pairs of components, the gradient is of a particular form. In other words, for a particular constraint equation, if the first component is i or the "head", and the second component is j or the "tail", the second component must always be the j to the first component's i. This head-tail relationship must remain consistent throughout the calculation of the constraint probabilities and their gradients.

For the spherical constraint probability (which is used by every joint), the gradient is found by

$$\begin{aligned}
&\nabla_{\theta_i} \ln P(c_s|\Theta) \\
&\quad = -\nabla_{\theta_i}(\mathbf{R}_i\mathbf{a}_{ij})\Sigma_s^{-1}(\mu_i + \mathbf{R}_i\mathbf{a}_{ij} - (\mu_j + \mathbf{R}_j\mathbf{a}_{ji})) \\
&\quad = -\begin{bmatrix} (e^{\hat{\mathbf{z}}\theta_{zi}} e^{\hat{\mathbf{y}}\theta_{yi}} \hat{\mathbf{x}} e^{\hat{\mathbf{x}}\theta_{xi}} \mathbf{a}_{ij})^\top \\ (e^{\hat{\mathbf{z}}\theta_{zi}} \hat{\mathbf{y}} e^{\hat{\mathbf{y}}\theta_{yi}} e^{\hat{\mathbf{x}}\theta_{xi}} \mathbf{a}_{ij})^\top \\ (\hat{\mathbf{z}} e^{\hat{\mathbf{z}}\theta_{zi}} e^{\hat{\mathbf{y}}\theta_{yi}} e^{\hat{\mathbf{x}}\theta_{xi}} \mathbf{a}_{ij})^\top \end{bmatrix} \Sigma_{c_s}^{-1} c_s \qquad \text{(II.22)} \\
&\nabla_{\theta_j} \ln P(c_s|\Theta) \\
&\quad = -\nabla_{\theta_j}(-\mathbf{R}_j\mathbf{a}_{ji})\Sigma_s^{-1}(\mu_i + \mathbf{R}_i\mathbf{a}_{ij} - (\mu_j + \mathbf{R}_j\mathbf{a}_{ji})) \\
&\quad = +\begin{bmatrix} (e^{\hat{\mathbf{z}}\theta_{zj}} e^{\hat{\mathbf{y}}\theta_{yj}} \hat{\mathbf{x}} e^{\hat{\mathbf{x}}\theta_{xj}} \mathbf{a}_{ji})^\top \\ (e^{\hat{\mathbf{z}}\theta_{zj}} \hat{\mathbf{y}} e^{\hat{\mathbf{y}}\theta_{yj}} e^{\hat{\mathbf{x}}\theta_{xj}} \mathbf{a}_{ji})^\top \\ (\hat{\mathbf{z}} e^{\hat{\mathbf{z}}\theta_{zj}} e^{\hat{\mathbf{y}}\theta_{yj}} e^{\hat{\mathbf{x}}\theta_{xj}} \mathbf{a}_{ji})^\top \end{bmatrix} \Sigma_{c_s}^{-1} c_s \qquad \text{(II.23)}
\end{aligned}$$

The gradient for the hardy-spicer joint is given by

$$\begin{aligned}
&\nabla_{\theta_i} \ln P(c_h|\Theta) \\
&\quad = -\nabla_{\theta_i}(\mathbf{q}_{ij}^\top \mathbf{R}_i^\top \mathbf{R}_j \mathbf{q}_{ji})\Sigma_h^{-1}(\mathbf{q}_{ij}^\top \mathbf{R}_i^\top \mathbf{R}_j \mathbf{q}_{ji}) \\
&\quad = -\begin{bmatrix} \mathbf{q}_{ji}^\top \mathbf{R}_j^\top e^{\hat{\mathbf{z}}\theta_{zi}} e^{\hat{\mathbf{y}}\theta_{yi}} \hat{\mathbf{x}} e^{\hat{\mathbf{x}}\theta_{xi}} \mathbf{q}_{ij} \\ \mathbf{q}_{ji}^\top \mathbf{R}_j^\top e^{\hat{\mathbf{z}}\theta_{zi}} \hat{\mathbf{y}} e^{\hat{\mathbf{y}}\theta_{yi}} e^{\hat{\mathbf{x}}\theta_{xi}} \mathbf{q}_{ij} \\ \mathbf{q}_{ji}^\top \mathbf{R}_j^\top \hat{\mathbf{z}} e^{\hat{\mathbf{z}}\theta_{zi}} e^{\hat{\mathbf{y}}\theta_{yi}} e^{\hat{\mathbf{x}}\theta_{xi}} \mathbf{q}_{ij} \end{bmatrix} \Sigma_{c_h}^{-1} c_h \\
&\nabla_{\theta_j} \ln P(c_h|\Theta) \\
&\quad = -\nabla_{\theta_j}(\mathbf{q}_{ij}^\top \mathbf{R}_i^\top \mathbf{R}_j \mathbf{q}_{ji})\Sigma_h^{-1}(\mathbf{q}_{ij}^\top \mathbf{R}_i^\top \mathbf{R}_j \mathbf{q}_{ji}) \\
&\quad = -\begin{bmatrix} \mathbf{q}_{ji}^\top (-\hat{\mathbf{x}}) e^{-\hat{\mathbf{x}}\theta_{xj}} e^{-\hat{\mathbf{y}}\theta_{yj}} e^{-\hat{\mathbf{z}}\theta_{zj}} \mathbf{R}_i \mathbf{q}_{ij} \\ \mathbf{q}_{ji}^\top e^{-\hat{\mathbf{x}}\theta_{xj}} (-\hat{\mathbf{y}}) e^{-\hat{\mathbf{y}}\theta_{yj}} e^{-\hat{\mathbf{z}}\theta_{zj}} \mathbf{R}_i \mathbf{q}_{ij} \\ \mathbf{q}_{ji}^\top e^{-\hat{\mathbf{x}}\theta_{xj}} e^{-\hat{\mathbf{y}}\theta_{yj}} (-\hat{\mathbf{z}}) e^{-\hat{\mathbf{z}}\theta_{zj}} \mathbf{R}_i \mathbf{q}_{ij} \end{bmatrix} \Sigma_{c_h}^{-1} c_h
\end{aligned}$$

(II.24)

The gradient for the elbow joint is given by

$$\begin{aligned}
&\nabla_{\theta_i} \ln P(c_e|\Theta) \\
&\quad = -\nabla_{\theta_i}(\mathbf{R}_i\mathbf{q}_{ij})\Sigma_e^{-1}(\mathbf{R}_i\mathbf{q}_{ij} - \mathbf{R}_j\mathbf{q}_{ji}) \\
&\quad = -\begin{bmatrix} (e^{\hat{\mathbf{z}}\theta_{zi}}e^{\hat{\mathbf{y}}\theta_{yi}}\hat{\mathbf{x}}e^{\hat{\mathbf{x}}\theta_{xi}}\mathbf{q}_{ij})^\top \\ (e^{\hat{\mathbf{z}}\theta_{zi}}\hat{\mathbf{y}}e^{\hat{\mathbf{y}}\theta_{yi}}e^{\hat{\mathbf{x}}\theta_{xi}}\mathbf{q}_{ij})^\top \\ (\hat{\mathbf{z}}e^{\hat{\mathbf{z}}\theta_{zi}}e^{\hat{\mathbf{y}}\theta_{yi}}e^{\hat{\mathbf{x}}\theta_{xi}}\mathbf{q}_{ij})^\top \end{bmatrix} \Sigma_{c_e}^{-1}c_e \qquad \text{(II.25)} \\
&\nabla_{\theta_j} \ln P(c_e|\Theta) \\
&\quad = -\nabla_{\theta_j}(-\mathbf{R}_j\mathbf{q}_{ji})\Sigma_e^{-1}(\mathbf{R}_i\mathbf{q}_{ij} - \mathbf{R}_j\mathbf{q}_{ji}) \\
&\quad = +\begin{bmatrix} (e^{\hat{\mathbf{z}}\theta_{zj}}e^{\hat{\mathbf{y}}\theta_{yj}}\hat{\mathbf{x}}e^{\hat{\mathbf{x}}\theta_{xj}}\mathbf{q}_{ji})^\top \\ (e^{\hat{\mathbf{z}}\theta_{zj}}\hat{\mathbf{y}}e^{\hat{\mathbf{y}}\theta_{yj}}e^{\hat{\mathbf{x}}\theta_{xj}}\mathbf{q}_{ji})^\top \\ (\hat{\mathbf{z}}e^{\hat{\mathbf{z}}\theta_{zj}}e^{\hat{\mathbf{y}}\theta_{yj}}e^{\hat{\mathbf{x}}\theta_{xj}}\mathbf{q}_{ji})^\top \end{bmatrix} \Sigma_{c_e}^{-1}c_e \qquad \text{(II.26)} \\
& \qquad \text{(II.27)}
\end{aligned}$$

II.5 Evaluation

To demonstrate the validity and generality of this approach, the proposed model is constructed for 2 types of articulated bodies: a 16 component, 15 joint hand, and a 10 component, 9 joint human body. The volumetric reconstructions were synthetically generated for the case of the hand, and generated using shape-from-silhouette using HumanEva II [27] image data in the case of the human body.

In both cases, ground-truth information about the position of the articulated body is available for comparison with the estimated results. The measures of accuracy used for both hand and human body test sequences is joint position error, proposed to be the standard measure of error [27]. Component (or segment) position and orientation error is also used for the hand case, for comparison.

For each test, the body model is manually sized and positioned near the actual voxel reconstruction of the body, and the parameters of the model mixtures is initialized accordingly. Fig. II.3 shows the model configured and result of the pose estimate in the first frame following initial model placement.

To generate the synthetic "voxel image" of the hand, each cylinder of voxels is positioned in the space described by the articulated body model. A sequence of 430 voxel images were generated of fingers bending from 0 to 90 degrees in a wave-like pattern while the palm also rotated. A few frames from the result of pose learning on this sequence are shown in fig. II.7. Note that the hand closes to a fist twice.

To measure its accuracy, the orientation and position of the individual segments of the estimated and ground-truth values are compared. Various statistics of the error were calculated, including mean absolute, root mean square, median, mode, and 95-th percentile. The histogram of component center

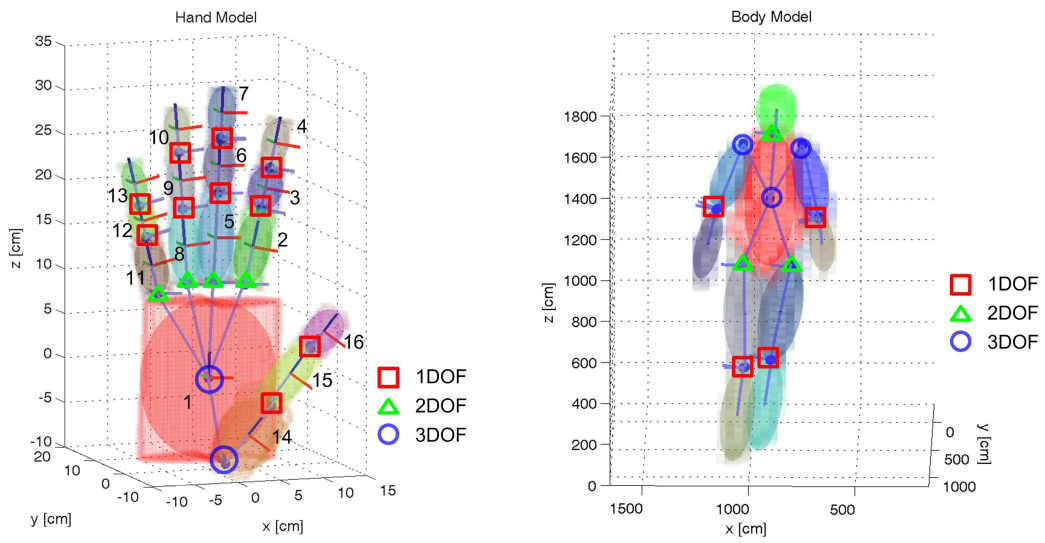


Figure II.3: Articulated body models in evaluation. Joint and joint types are annotated according to the number of degrees of freedom. The center of the palm and torso are used as the center of the body and carry an additional 3 degrees of freedom for translational movement. Dimensions are based on actual bodies.

position and angular orientation error over all components are illustrated in fig. II.4. The same statistics by component is illustrated in fig. II.5(a). With a voxel resolution of 0.5 cm to the side of each voxel, component center positional accuracy of 0.33 cm mean absolute error could be achieved with the hand sequence. Mean angular error measured 8.51°. Using the joint-position error metric, mean absolute error yields 0.5 cm joint position error. Fig. II.5(b) shows the statistics of this error overall and by component. The dimensions are based on an actual hand, so the results from this test are representative of the accuracy of learning hand pose with ideal reconstructed volumes.

The next data-set used to test the algorithm is the HumanEva II data-set [27]. Subject 2 and 4 were both used. Shape-from-silhouette is used to generate the voxel reconstruction of the human subject. Silhouettes are generated by background subtraction in the HSI color space, followed by connected component analysis, retaining only the largest components.

The performance measures consist of absolute and relative 3D spatial error of joint locations using mean Euclidean distance. Relative 3D joint location error is calculated relative to the torso point. All joints as prescribed in [27] were utilized in the error measurement. The results are tabulated in tab. II.2. The mean joint position error over time and histogram plots are shown in fig. II.6(a) and II.6(b).

The body model is sized and positioned over the voxel reconstruction for the first frame, and then the algorithm is allowed to process the remaining frames. Three segments of the S2 sequence were processed separately: 1–190, 250–500, and 700–1220. The entire S4 sequence was processed in a single

run. Several frames from the pose learning results in S2 are illustrated in fig. II.8.

The S2 sequence was processed in 3 segments because of loss of tracking during frames 190–250 and 500–700. Within these ranges, the subject was turning at a particular position in the space such that the voxel reconstruction produced a diamond shaped reconstruction as observed from the top. This particular camera configuration is unable to carve away the erroneous volume in this situation. This causes the algorithm to fall into an erroneous local maximum of a pose that is turned 90 degrees from the correct pose along the length of the body. Just as the EM-algorithm is subject to convergence in local-maxima, this algorithm is no different and the algorithm does not recover from this. One should keep in mind that these results are from a generic pose learning algorithm that uses only volume information, albeit the silhouettes are derived from imagery. This problem is left to be investigated in the future.

In sequence S4, this phenomenon can be seen between frames 350–600 when the error jumps from 17 cm to 35 cm. During this period, the algorithm is tracking the body with the model pose reversed (left is right and vice versa). As subject 4 walked around the second time, the model reversed a second time and then remained in the correct direction for processing for the remaining frames. We sudden spike that extends to 1.2 meters starting at around frame 300 is the result of an error from the ground-truth.

The primary reasons for the large differences in accuracy between the hand and human body test sequences are the discrepancies between the dimensions of the human body in the model and reality, and the quality of the voxel reconstruction. The human body model used for this test was created by placing the model components in the voxel reconstruction, part by part. This implies that the joint locations are only approximately in the position of the true joint locations, resulting in the consistently greater than 8cm error in both subjects. As compared to the hand model, the hand model itself was used to generate the voxels and little discrepancy resulted.

The loss of track as well as diminished accuracy in the human body test sequence can also be attributed to the quality of the voxel reconstruction. While the hand sequence can be considered ideal voxel reconstructions, the human body voxel reconstructions are limited by several factors. Finite number of cameras and the given camera configuration results in poorer voxel reconstruction in some regions of scene compared to others, as described above with the diamond shape reconstruction. This is a limitation in shape-from-silhouette [7, 8]. Another source of error is in segmentation. Although shadows were mostly eliminated by using the HSI color space, dark areas of the subject in the scene looked very similar to shadow and was sometimes excluded in the silhouette, carving out valid areas in the voxel reconstruction.

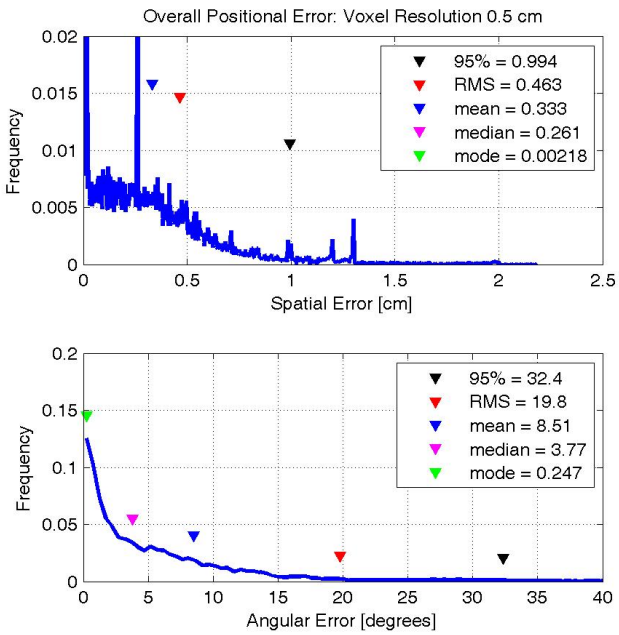


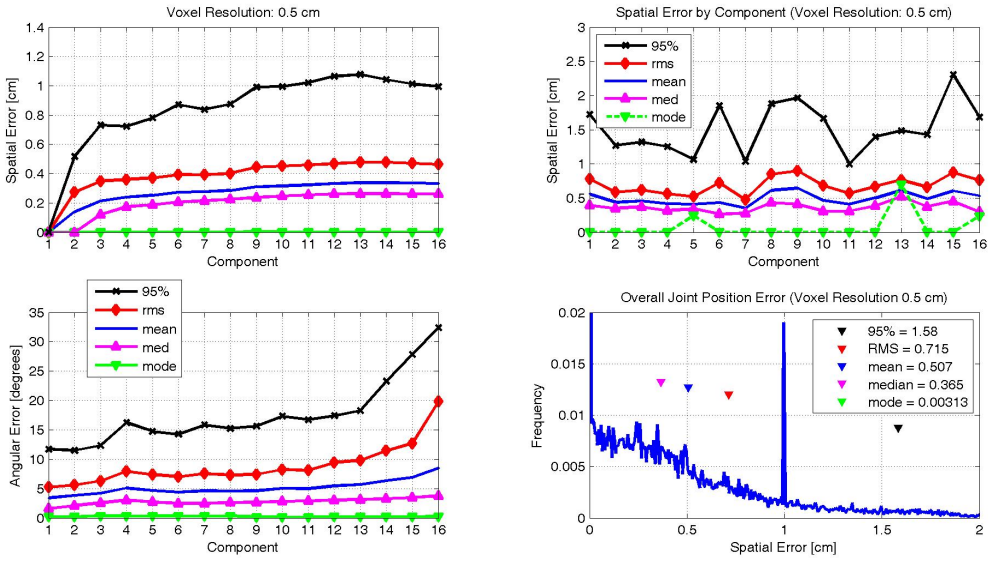
Figure II.4: KC-GMM Estimation Error: Histogram of component center and angular orientation estimation error with respect to ground-truth for synthetic hand data.

II.6 Discussion and Concluding Remarks

Because the model represents an articulated body model that only encourages configurations of Gaussian components where the joints will have the specified 1, 2, and 3 DOF, limbs bending in infeasible directions are possible solutions in the learning process. Additional constraints are required to limit the range of motion in the joints. This will also improve the tracking performance from one frame to the next by eliminating some erroneous local optima in the pose space.

The model currently does not contain a mechanism to utilize temporal information from the last processed frame, such as velocity, with which this system will benefit. The results shown merely utilize the last estimate as a starting point for learning the pose.

Despite both these shortcomings, it is clear that volumetric data alone is sufficient to recover the pose of a hand even through a nearly closed fist. The model is general enough to be easily extended for other articulated bodies. The primary contribution of this chapter is a kinematically constrained Gaussian mixture model that relates volume data and the pose of the articulated body and the means to learn the pose using the EM algorithm; no additional constraint optimization steps external to the EM algorithm are required. The algorithm was validated on two types of articulated bodies.



(a) Mode, median, mean, RMS, and 95th-percentile statistics of component center and angular orientation error with respect to ground-truth from synthetic hand data.

(b) Overall and by-component *joint* position error with respect to ground-truth on synthetic hand data. 3-D joint position error, as opposed to component center and orientation error, is the proposed standard estimation error measures for the HumanEvaII data set.

Figure II.5: KC-GMM Estimation Error: Error by component.

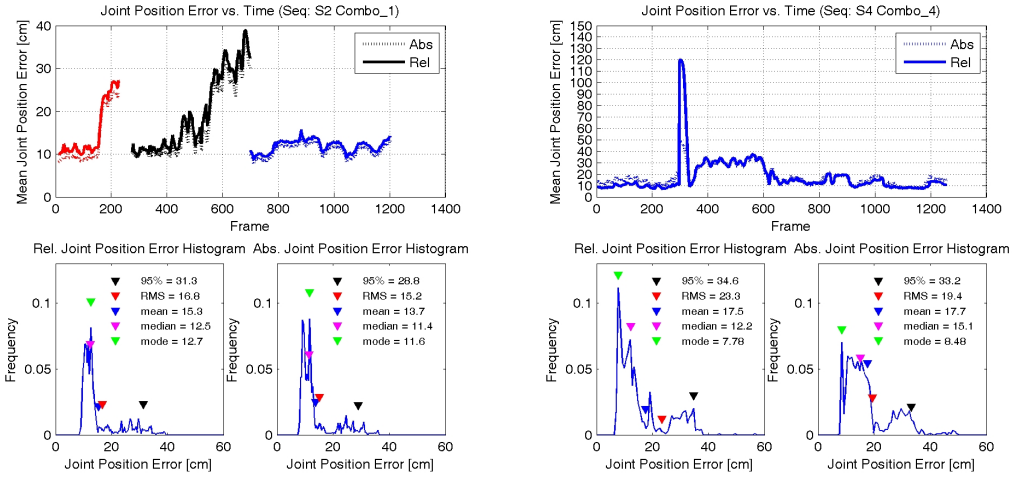
II.7 Future Work

Many aspects of the work presented in this chapter can be extended. Several directions are presented in the following.

II.7.A Body Structure Learning.

Work in variational approximation for inferring latent variables in hierarchical graphical models, such as kc-gmm, can potentially be applied directly to learning the number of components of this model, namely the number and arrangement of Gaussian components [28].

We believe that the proposed method of constraining the mixture components is one module in a complete articulated body *structure* and pose learning algorithm. Using Gaussian mixtures as the basis of representing volumes, it is conceivable several smaller components can represent a single rigid body, akin to several atoms making up the larger whole. A path for further investigation can be to augment the model to constrain the range of motion, and incorporate temporal cues. Eventually, the investigation can lead to a structure learning computational framework that begins with composing a volume with several small Gaussian components, and components that move together over several frames can meld together



(a) Subject 2 Combo Trial 1

(b) Subject 4 Combo Trial 4

Figure II.6: Overall joint position error from ground-truth of articulated body pose learning on HumanEvaII human body data set.

into larger components or components connected by joints constrained by the 1, 2, and 3 DOF joint constraints as described here.

II.7.B Joint Angle Constraints and Temporal Information.

The current model does not contain a mechanism to constrain the range of motion of the joints. This may or may not be considered a drawback depending on the application. It has been argued that when measuring, for example, abduction of knee for knee injury analysis, one would prefer a motion capture system that is able to allow all directions of movement of the model for that joint. For most other applications however, such additional degrees-of-freedom limits the algorithm's robustness.

II.7.C Alternative Volumetric Reconstruction Techniques.

Shape-from-silhouette was used to generate the visual-hull reconstructions used in this algorithm. SFS makes no use of color information beyond the background-subtraction algorithm to generate the silhouette. It is possible to utilize color information to refine the voxel reconstruction further, creating tighter upper-bounds on the actual volume and avoiding holes in the reconstruction. For example using the photo-consistency constraint, one can arrive at the so-called photo-hull, which is a tighter upper-bound than the visual-hull. In recent years, there has been a flurry of new approaches to extracting 3-D reconstructions. A thorough consideration of these approaches should reveal an appropriate method for generating data for body pose tracking in the vehicle and in other environments [7, 29].

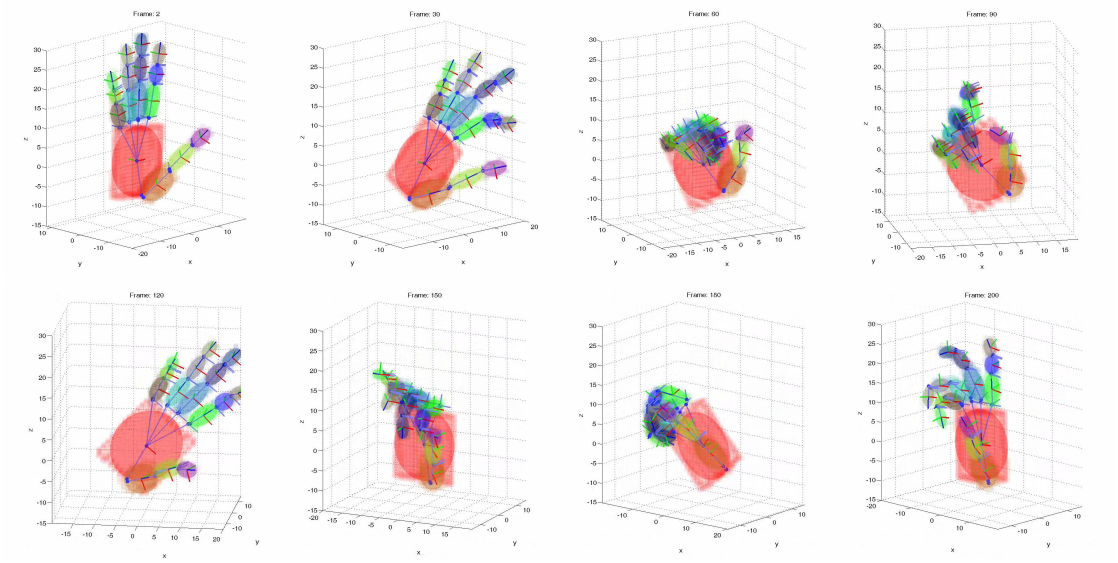


Figure II.7: Hand pose learning results on synthesized hand volume reconstructions of a hand moving its fingers in a wave-like pattern while rotating at the palm.

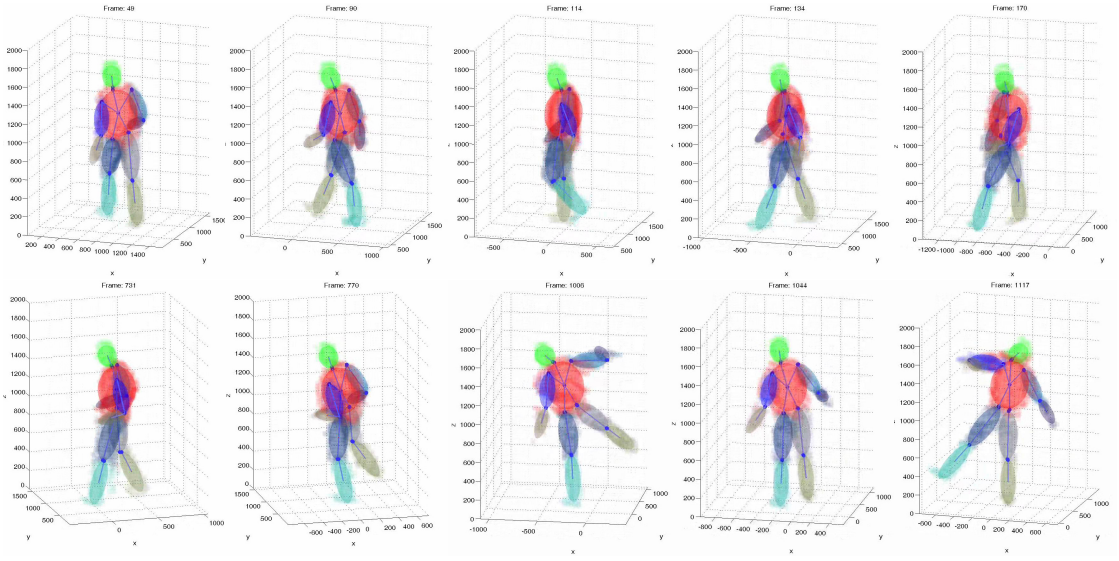


Figure II.8: Body pose learning results on actual image data of a human subject walking, running and balancing. Frames 49, 90, 114, 134, 170, 731, 770, 1006, 1044, and 1117 are shown.

Table II.2: Joint position error summary for kc-gmm pose learning on the HumanEva II data-set. First group follows the prescribed standard evaluation. Second group includes only successfully tracked frames.

			Rel. 3D Error			Abs. 3D Error		
		Frames	Mean	RMS	95-th	Mean	RMS	95-th
Prescribed Subsequences	S2 Combo_1: W	1–350	14.1	15.2	26.6	12.5	13.7	24.2
	S2 Combo_1: WR	1–700	17.9	19.7	33.6	16.0	17.8	30.7
	S2 Combo_1: WRB	1-1202	15.3	16.8	31.5	13.7	15.2	28.7
	S4 Combo_4: W	1–350	17.1	30.8	102	16.1	18.1	42.7
	S4 Combo_4: WR	1–700	21.8	29.2	36.8	21.0	22.9	34.7
	S4 Combo_4: WRB	1-1220	17.7	23.6	34.5	17.7	19.6	33.3
Successfully Tracked Subsequences	S2 Combo_1:W	1–160	11.0	11.1	12.3	9.2	9.3	10.3
	S2 Combo_1:W	275–350	10.8	10.8	11.8	10.2	10.3	11.2
	S2 Combo_1:R	350–550	13.5	13.8	19.0	11.8	12.0	16.5
	S2 Combo_1:B	700–1202	12.4	12.0	13.8	10.9	11.0	12.8
	S4 Combo_4:W	1–350	14.1	15.2	26.6	12.5	13.7	24.2
	S4 Combo_4:R	700–1258	12.1	12.6	19.2	13.4	13.9	18.8
	Average		**15.9**	**17.9**	**27.7**	**11.9**	**12.3**	**16.9**

Sequences: W-Walking, R-Running, B-Balancing
All units are in cm.

The text of Chapter II, in part, is a reprint of the material as it appears in: Shinko Y. Cheng and Mohan M. Trivedi, "Articulated Human Body Pose Inference from Voxel Data Using a Kinematically Constrained Gaussian Mixture Model," in Proceedings and best paper award winner of the Workshop on Evaluation of Articulated Human Motion and Pose Estimation in conjunction with IEEE CVPR, 2007, and Shinko Y. Cheng, Mohan M. Trivedi, "Multimodal Voxelization and Kinematically Constrained Gaussian Mixture Model for Full Hand Pose Estimation: An Integrated Systems Approach," in Proceedings of IEEE International Conference on Computer Vision Systems, Jan. 2006, pages 34-42. I was the primary researcher of the cited materials and the co-author listed in these publications directed and supervised the research which forms the basis of this chapter.

III

Driver Intersection Turn Intent Inference

In this chapter we attempt to shed light on how and to what extent driver gesture indicates driver intention to perform a driving maneuver. We first present our studies on the intersection-turn maneuver and motivations for using body pose to determine its onset. We also introduce an algorithm for recognizing driver intersection-turn intent, and present an analysis of the extent that driver body part information contributes to the detection of this intent.

III.1 Introduction

There are four objectives of this chapter. The first is to present a characterization of the problem of driver intent recognition, and introduce the "slow" intersection turn maneuver, the predominant type of turn maneuver among intersection approaches. Second, we present a new system for recognizing driver turn-intent before the vehicle enters the intersection by utilizing information about the driver and vehicle dynamics. Third, we present our findings on the most appropriate performance measures for evaluating driver intent inferencing systems, beyond receiver-operator-characteristics (ROC) curves. Finally, we present our findings on the added value of possessing body pose information using the proposed system and show that derived body pose information in the form of steering angle, throttle, and brake activation together with other vehicle dynamics provide the set of cues for maximal intent recognition rates.

Our contributions of this research extend to the following 4 areas.

1. We present a characterization of intersection approaches, including intersection turns, by observing data collected along a 4-hour-long drive through 258 intersections. Among the types of intersection approaches, we identified "slow" intersection-turn maneuvers to have very consistent attributes

representing 93.7% of all turns in our data-set. We present a study of trends in body movement and vehicle dynamics during the course of this "slow" intersection turn maneuver.

2. We introduce an integrated system for recognizing intersection-turn intent utilizing a) a set of vehicle and driver body dynamics information, b) a series of pre-processing steps for the creation of the feature vectors to yield empirically optimal recognition rates, and c) the kernel Relevance Vector Machine (RVM) as the pattern classifier.
3. We present the experiment design and evaluation of the driver intent recognition system using vehicle dynamics information and high-fidelity driver body movement information from a novel input modality. The evaluation considers ROC curves which measure performance over instances, and event-aligned response-statistics plots to examine classifier behavior over time.
4. In the process, we also examine the extent to which driver head and hand pose information of that fidelity contributes to improved driver intent inference.

This chapter is organized as follows: We will first present related work on driver intent recognition and the optical motion capture system as a primary source of pose data in sec. III.2. We then examine the types of intersection approaches that exist from the data that was collected in sec. III.3. We then present the intersection turn intent system in sec. III.4. Finally, we will evaluate the system using both the ROC curve and the proposed statistical time-aligned response plots in sec. III.5.

III.2 Related Work

The idea of using body pose as a cue for analyzing driver maneuvering behavior only recently appeared in literature. Much of the effort has been in designing systems that predict driver intent to perform a lane change [30–34] which was identified as one of the more dangerous driving maneuvers [4]. A number of works also looked at intersection-turn maneuvers [32, 33] which account for 27% of U.S. traffic collisions [35]. We describe 5 results in detail that are most related to ours. Each description is followed with a description of our contribution to the discussion of driver maneuver intent recognition.

The work from Kuge *et al.* [34] and Pentland and Liu [33] proposed modeling human behavior using the hidden-Markov model or extension. Kuge *et al.* [34] proposed using HMMs to model the time-series data collected during the lane-change. They focused on the lane-change maneuver, and on modeling the steering angle and angular velocity modality during the maneuver. Their model was able to distinguish between a normal lane change, an emergency lane change and lane keeping maneuver 100% of the time (with a false alarm rate of 0.29% for the emergency case) about .7s from its onset. The data was collected from subjects driving in a simulator.

Pentland and Liu proposed a more complicated model referred to as the Markov Dynamic Model. It consists of a lower-level dynamical model (HMM) describing small-scale prototypical human behavior capable of modeling the smoothness or temporal correlation of the measured data, and

an upper-level model describing large-scale structure by coupling together these lower-level states into a Markov chain. The results were very good using, again, simulation data with a true-positive rate of $95.24\% \pm 3\%$ at 1.5 seconds from the onset of passing, intersection-turning and lane changing maneuvers. With regards to intersection-turns, entering the intersection was assumed to have always taken place at 2.5 seconds from the start of the turn, which meant their model supposedly predicted the onset of a turn 1 second before the vehicle entered the intersection. It was difficult to read what was the associated false alarm rate during the times when the driver was performing a different maneuver.

Both of these approaches modeled the maneuver from the point of view of the driver; the start of the maneuver is when the driver consciously begins the maneuver. We define the starting point of the intersection-turn at the moment when the vehicle enters the intersection. This has, as we will show, allowed a different interpretation of recognition performance by defining it as the proportion of maneuvers predicted before the vehicle enters the intersection, as opposed to the proportion recognized within some seconds from the onset of the maneuver. These two ideas are not mutually exclusive. Our definition is the cleanest definition to fulfill a requirement of defining the feature-vector for the classifier. From the point of view of evaluation, the number of turn-intents correctly recognized is just as important to report *explicitly* as the proportion recognized correctly t seconds from maneuver onset since the real danger is when the vehicle is in the intersection.

Another point of contrast is their use of driving data from a driving simulator; our experiments are based on data collected from driving on real streets. As a result, our performance measurements are representative of actual performance when driving on real roads. We do acknowledge as they have pointed out the need for time-warping, the model attribute that is able to take into account the different speeds at which the maneuvers take place; but this can usually be avoided by training with more examples [36].

Oliver *et al.* [32] also proposed using HMMs to infer 7 different driving intentions including intersection-turn intent. Data was collected from driving on actual streets. Input cues considered came from 8 sources including driver pose information: they were speed, throttle, brake activation, gear, steering angle, gaze (depicted as 6 discrete states), relative obstacle speed and relative lane position. The 7 driving behaviors were passing, turning left and right, changing lanes left and right, starting and stopping. They report predictive power of their models to be on average 1 second before "any significant" change took place in car or contextual signals. The true-positive rates were 85.7% and 66.7% for the left and right turns, respectively.

False alarm rates were not reported with these true-positive rates however. False alarm rates are a very important performance statistic to report, because an insufficiently low rate would doom the method to rejection by the consumer. In the discussion of human-machine interfaces for the new semi or fully automated driver-assistance systems, such as headway distance control or lane keeping control, achieving smooth control mode transitions from automated to manual operation is of paramount importance [32,34].

High false alarm rates would introduce feelings of incongruity in ordinary driving.

Oliver *et al.*also consider the starting time of the intersection turn from the point-of-view of the driver. Again, the difficulty here is determining where the vehicle is at 20% into the maneuver. We propose the use of a spatial landmark as the alignment point for reporting intersection-turn intent recognition rates.

Salvucci *et al.* [30, 31] suggest a totally different approach by inferring driver intent to make lane changes using model-tracing, a knowledge-based technique with roots in education and efforts in enabling tutoring systems to predict what steps the student intends to do next in problem solving for online interactive help. This technique involves comparing the desired steering angle and acceleration parameters over time with several simulated parameter trajectories at various stages of maneuver completion to determine whether or not the driver is performing a lane change maneuver. The lane change start is defined when the vehicle achieves the minimum lateral velocity and proceeds without lateral reversal through the lane boundary into the destination lane. This research was the first to consider sample-by-sample performance of lane change behavior recognition. The results from their experiments were 82% detection rate within 0.5s from the onset of the maneuver, and 93% within 1s, both at 5% false alarm. Experiments were performed in both a simulator and real vehicle.

They also claim that the same approach can be used to recognize intersection-turns, although it was reserved for future work. One aspect different than previous work has been their analysis of spatial positioning of the vehicle at t seconds from the onset of maneuver. In this case, they reported average time to lane crossing enabling an understanding of the percentage of intents that were recognized before the vehicle crossed the lane boundary. We will present analysis on intersection-turn intent that will also provide an understanding of the algorithm's sample-by-sample performance.

McCall *et al.* [36] proposed to use the kernel-RVM model under the classification framework in recognizing lane-change intent. The algorithm detects driver intent by recognizing preparatory motion of the driver's head, together with the vehicle's dynamics information. As a result, they report that an additional 0.5 seconds lead-time could be gained for the same recognition performance without head motion data. The performance numbers of their approach were as promising as the other approaches, but the results were evaluated with the receiver-operator-characteristic plot, and trained with data from a real vehicle. They report 95% detection rate at 5% false alarm for a prediction time of 2.5s before lane crossing. These results were the main inspiration of our proposed intersection-turn intent inference algorithm.

In contrast to the work of McCall *et al.*, we introduce two significant differences: 1) We focus on the intersection-turn maneuver intent. The non-trivial aspect is defining the intersection-turn "event" as this classification approach requires. We collected over 250 intersection-turn encounters and describe a systematic categorization of the various types one would encounter. 2) We also explore the use of very high-fidelity head and hand pose information for use in recognizing any preparatory motion that may exist

moments before the intersection-turn. As a minor addition, we also propose the use of a new visualization of the results to gauge the classifier's performance over time or sample-by-sample performance. We refer to it as the time-aligned statistical response plot.

III.2.A Driver body-pose analysis

Driver body pose is a key element in determining the extent to which body pose contributes to the reliable recognition of driver maneuver intent. We examined a variety of existing motion-capture technologies that can recover body pose: including mechanical, magnetic, and optical marker-based systems. These motion capture systems provide both 3 degrees-of-freedom position information of a point on the subjects' body, and full 6 DOF body-part position and orientation. However, these techniques require that the subject wear something during the capture, either the sensor device itself or markers that are tracked from sensors placed some distance away.

For this reason, image-based motion capture is the more appropriate approach in a consumer car, because it does not require special markers or user intervention. For more than 25 years, the computer vision community has researched image-based body-pose recovery [10], developing algorithms that rely only on passively sensed data of the subject to analyze and extract body-pose information.

However, vision-based body-pose recovery is complicated by the vehicular requirement for algorithms to be robust to changing illumination. We recently showed (and will present in chap IV) the promise of using long-wavelength infrared imaging sensors to locate exposed skin and recover hand position and pose in the presence of lighting that changes from one moment to the next [37]. This approach senses not the visible appearance but the thermal response of the hands, which hovers around a constant temperature. However, this approach must also contend with varying temperatures inside the car—a much slower source of variation in the scene—and the cost effectiveness of using multiple thermal infrared cameras.

Consequently, we rely on a marker-based method of recovering poses. Such a method, if fast and accurate, can help our objectives in two ways. First, it can provide ground-truth data for pose estimation development. As demonstrated in Chapter II, this capability is critical in gauging a vision-based pose-estimation systems performance among techniques. Second, it helps us understand the upper limits of an activity-recognition system based on pose data by providing the system with the cleanest, most complete data-set. Knowing this limit helps justify dedicated development efforts toward passive and likely application-specific pose-recovery systems.

Here, we focus on the second benefit which is clean data. Using driver body-pose information collected from marker-based motion capture, we developed a system to recognize and predict driver left and right intersection-turn behaviors, and analyze the extent to which body pose, in the form of head and hand position, contributes to that task.

Table III.1: Related work on driver intent analysis and recognition.

	Objective	Gestural Input Considered	Other Input Considered	Method	Results TP(FP)	Comments
Kuge *et al.*('98) [34]	Lane Change Intent (emergency & normal)	steering angle, steering angular velocity	n/a	HMM	100%, 100% at .7s from onset	- Driving data from simulation only. - Models behavior from start of maneuver. - False positive rate for emergency LC reported only (0.29%).
Pentland & Liu ('99) [33]	Turning Intent Passing Intent Lane Change Intent Stopping Intent	steering, throttle	speed	Markov Dynamic Model (2-level HMM)	$95.24\% \pm 3\%$ at 1.5 seconds from onset	- Driving data from simulation only. - Models behavior from start of maneuver.
Oliver & Pentland ('00) [32]	Turning Intent Passing Intent Lane Change Intent Stopping Intent Starting Intent	steering, throttle, brake, gaze	speed, surrounding obstacle dynamics, gear, lane position,	HMM, CHMM (PIN,DAG)	85.7%, 66.7% by .8s and .5s from onset	- Driving data from simulation and real vehicle. - Models behavior from start of maneuver. - False positive rate not reported.
Salvucci *et al.*('04, '07) [30, 31]	Lane Change Intent	steering, throttle	headway, obstacle dynamics, # of lanes, lane pos., lane deviation at 10m/30m ahead	Model tracing	82% (5%) within 0.5s of onset	- Driving data from real vehicle. - Models behavior from start of maneuver.
McCall *et al.*('05–'06) [36]	Lane Change Intent Brake Intent	steering, throttle, Head motion, Foot hover	speed, headway, lane position, yaw rate	kernel-RVM	95% (5%) 2.5s before lane crossing 60% (5%) 1s before braking event	- Models behavior from spatial event (lane crossing).

Continued on next page

Table III.1 – continued from previous page

	Objective	Gestural Input	Other Input	Method	Results	Comments
Cheng *et al.*('05) [37]	Turning Intent	brake, throttle, steering, head motion, hand position, steering wheel grasp	speed	HMM	100% left, 100% right, post-fact with steering only	- Models behavior from spatial event (entering intersection). - False positive rate not reported. Percentage recognized at onset of maneuver not reported.
This Approach	Turn Intent	brake, throttle, steering, **head pose, hand position, turn-signals**	speed	**RBF kernel-RVM**	86%(5%) left, 74%(5%) right (w/o turn-signals) at moment of entering intersection	- Models behavior from spatial event (entering intersection). - False positive rate not reported. - Percentage recognized at onset of maneuver not reported.

III.3 Types of Turning Maneuvers

We characterize an intersection turn as one that requires a change in vehicle heading of approximately 90 degrees. A driver would encounter such turns when maneuvering a vehicle at T and four-way intersections, as well as turnouts into side streets. These intersections have various numbers of start and end lanes, combinations of one- and two-way streets, and sometimes certain traffic controls (such as stoplights, stop signs, or yield signs).

There are many other kinds of turns–those involving freeway ramps or curves in the road, those made to avoid road obstructions, U-turns, and so forth. We focus on intersection turns because they are the most prevalent in urban driving (an observation made from our data-set) and are most prone to involve road obstacles. We identified two kinds of intersection approaches: 1) the driver can stop, wait at least two seconds, and then start into the turn (a halt turn), or 2) the driver can stop for less than two seconds and then start into the turn, or merely slow down before entering into the turn (a slow turn). We focus on the first kind by specifically training our system to ignore all other kinds of turns and behaviors. The slow-turn represents 93.7% of all turns in our data-set.

The vehicle dynamics and driver body pose data was collected from the LISA-P test-bed. The data consists of 4 hours of natural driving (from a single driver), containing 258 intersection encounters. After the collection, the type of traffic controls at each intersection was recorded. The stop sign, yield sign and traffic light controls were observed. The types of intersections encountered were cross (4-way) intersection, t-intersection, curve of approximately 90° along the road, and end of the road. Tab III.2 summarizes the types of each intersection that were encountered.

Table III.2: Summary of Intersection Attributes.

(a) Traffic controls

	#	%
None	120	46.5
Light	93	36.0
Stop-sign	44	17.1
Yield-sign	1	0.4
Total	258	100.0

(b) Intersection types

	#	%
Curve	123	47.7
T	99	38.4
Cross (4-way)	33	12.8
End	3	1.2
Total	258	100.0

The start and end times of each intersection encounter were recorded. These times indicate when the nose of the vehicle crosses the boundaries between the road and the intersection. Of all 258 encounters, each approach was categorized as one of the following:

1. Slow approach - Approach the intersection by stopping completely for less than 2 seconds (including 0 seconds or just slowing down), then proceed into the intersection.
2. Halting approach - Approach the intersection by stopping completely for 2 or more seconds, then proceed into the intersection.

3. Fast approach - Approach the intersection without activating the brake prior entering the intersection.

As indicated in sec. III.3, when the intersection turn-maneuver is categorized according to the slow, halting and fast approaches, we can observe that a vast majority of the turning was of the slow kind comprising 93.7% of all turns. A total of 159 slow turns were recorded in the data-set, which represents 61% of all intersection encounters. Each instance serves as a positive example in the data-set. Tab. III.3 summarizes the number of instances of each type of intersection approach encountered and the timings of the slow turn.

Table III.3: Collected Intersection-turn Durations.

(a) Types of Intersection Approaches

Turn type	#	%	% of all turns
Slow go straight	15	5.8	-
Slow turn left	74	28.7	42.5
Slow turn right	85	32.9	48.9
Slow U-turn	4	1.6	2.3
Halt go straight	3	1.2	-
Halt turn left	9	3.5	5.2
Halt turn right	2	0.8	1.2
Fast go-straight	63	24.4	-
Total	**255**	**100.0**	**68.2%(174/258)**

(b) Slow Turn Durations

	Left turn	Right turn
Minimum (sec)	2.14	2.21
Average (sec)	5.89	4.06
Maximum (sec)	9.95	7.30
Standard deviation (sec)	1.59	1.24
Total number of turns	74	85

The patterns of driver body pose during these kinds of intersection turn approaches were examined, and there appeared to be distinct patterns which could be machine recognizable. These patterns can be observed when the pose sequence of each instance of an intersection turn are plotted on top of one another, time-aligned to the start of the intersection turn $t = 0$. Fig. III.1 and III.2 show the overlaid head and hand pose parameter sequences. For a complete list of cues that were captured, see sec. III.4. Distinct patterns can be observed between -2s and 4s surrounding the start of the turn. The warmer colors indicate that many instances follow that particular pattern over time. The color values are normalized by the maximum value encountered for each column of bins.

Between the sets of input cues depicted in the two figures, fig. III.1 shows the most distinct patterns through the intersection turn. One can clearly see in these plots that the head moves with the turn maneuver in the corresponding direction. The pattern is seen to start at -1s for the left and -2s for the right

turn, and last up to 4s after the turn started. The head was not observed to translate (move tangentially) in any strong pattern. There is a pattern of head movement in the z-axis (relative to the first camera in the set-up) for the left turn between -6s and 0s when the head moves nearly 2 cm in the negative z-direction, which in terms of long-term predictability is a favorable observation.

Fig. III.2 shows the patterns of the hand positions. Here, a lot of movement of the hands can be observed during the maneuver. It is very apparent that the hands move in a certain way through the turn. However, there are large variations in hand movements over all instances of the turn. Looking at "left hand y", we can see that the hand sometimes begins movement at -3s, and other times only after 0s, the start of the turn. An ascent however is always present, and it is up to the model to capture this pattern.

Another interesting observation is that the right hand is shown to move in two distinct ways for the left and right turns. This could be seen in "right hand z" where there are two hot spots during the duration of the intersection turn. This is an artifact of the data collection system. Although precautions were taken to distinguish the left from the right hand with different marker configurations, there were several instances of maneuvers when the left hand measurement was actually that of the right hand and vice versa. Fortunately, this switching occurred only when the hands occluded each other after the turn had begun, maintaining a certain consistency of the tracking before the start of the intersection maneuver. This may have ultimately caused a slight degradation in recognition performance.

Vehicle dynamics cues are also visualized the same way in fig. III.3 and III.4. Again between -2s and up to 4 seconds, there appear to be distinct patterns associated with the intersection turn. One can see in fig. III.3 that both the throttle and torque produced by the engine dip to 0% as the driver approaches the turn lasting between -2s and 0s. After entering the intersection, the vehicle begins to accelerate as expected into the turn. The speed also exhibits a pattern through the turn as well, although not as tightly correlated. The speed dips between -2s and begins to accelerate again after 0s. The brake is almost always activated 2s before the turn, and released just after entering the intersection. We can also see steering values change towards the direction of the turn, just like "head ty". The change begins at around -1s , lasting to 4 or 6s after the start of the turn. Steering angular velocity exhibits a pattern over about the same duration.

To conclude this section, patterns appeared to exist in both the body pose and vehicle dynamics cues. Body pose exhibited some semblance of a pattern to be recognized up to -6s from the start of the turn. The next sections describe the challenge of devising an appropriate model to capture this pattern, along with classifier training results using these input cues.

III.4 Driver Intersection-Turn Intent Recognition

Following initial capture of sensor data, the driver intersection turn intent inferencing system consists of a feature vector extraction stage and a pattern classification stage. The objective of the feature extraction stage is to represent the raw sensor data in the form that emphasizes the discriminating patterns of the data. The flow diagram consists of 1) sensors extracting body pose and vehicle dynamics information, 2) a pre-processing step to construct the feature vector, and 3) the pattern recognizer taking the feature vector as input and providing a probability of an impending turn maneuver as output. Fig. III.5 shows the flow diagram of the system.

Input data to the prediction system is collected from sensors aboard the LISA-P test vehicle. The process of capturing these vehicle dynamics attributes is described in app. B. The vehicle dynamics of relevance to the intent prediction system include the following attributes:

Vehicle Dynamics Attributes [units]:

1. Steering angle [degree]
2. Steering angle speed [degree/s]
3. Vehicle speed [km/h]
4. Brake activation [on/off]
5. Engine Torque [%]
6. Throttle [%]
7. Left Turn-signals [on/off]
8. Right Turn-signals [on/off]

In addition to the vehicle dynamics, body part pose position and orientation are also captured. Specifically, the 6-degree-of-freedom (DOF) head pose and the 3-DOF hand positions are captured using a retro-reflective marker based motion capture system. Altogether 12 parameters describe the pose of the driver.

Body Pose Attributes [units]:

1. Driver Head Position (x, y, z) [mm]
2. Driver Head Orientation $(\theta_x, \theta_y, \theta_z)$ [radians]
3. Driver Left Hand Position (x, y, z) [mm]
4. Driver Right Hand Position (x, y, z) [mm]

The captured time-series are accumulated into windows of M-sample length sequences.

$$\mathbf{Y}'_n = \begin{pmatrix} \mathbf{H}'_n \\ \mathbf{L}'_n \\ \mathbf{R}'_n \\ \mathbf{C}'_n \end{pmatrix} = \begin{pmatrix} \mathbf{h}_n & \mathbf{h}_{n-1} & \dots & \mathbf{h}_{n-M+1} \\ \mathbf{l}_n & \mathbf{l}_{n-1} & \dots & \mathbf{l}_{n-M+1} \\ \mathbf{r}_n & \mathbf{r}_{n-1} & \dots & \mathbf{r}_{n-M+1} \\ \mathbf{c}_n & \mathbf{c}_{n-1} & \dots & \mathbf{c}_{n-M+1} \end{pmatrix} \in \mathbb{R}^{D \times M} \tag{III.1}$$

Each sequence consists of time-synchronized samples of the sensor input. Here, $\mathbf{h}_n \in \mathbb{R}^6$ is the 6-DOF pose (position and orientation) of the head, $\mathbf{l}_n, \mathbf{r}_n \in \mathbb{R}^3$ is the 3-DOF position of the left and right hands, and $\mathbf{c}_n \in \mathbb{R}^8$ is the set of vehicle dynamics information, all at time n.

Each M-sample input sequence is then sub-sampled by a rate of S by summing over time every S samples in the time-series. The resulting sub-sampled sequence is $L = M/S$ in length. The value of M is chosen such that M is divisible by the integer value S. If we let $\{\mathbf{H}_n, \mathbf{L}_n, \mathbf{R}_n, \mathbf{C}_n\}$ be the set of sub-sampled L length sequences, we can then construct the un-normalized feature vector

$$\mathbf{y}_n = \text{vec}\left(\mathbf{Y}_n\right) = \text{vec}\begin{pmatrix} \mathbf{H}_n \\ \mathbf{L}_n \\ \mathbf{B}_n \\ \mathbf{C}_n \end{pmatrix}, \qquad \mathbf{Y}_n \in \mathbb{R}^{D \times L} \tag{III.2}$$

where vec$(\cdot)$ is the vectorization operation.

This un-normalized feature vector $\mathbf{y}_n$ is finally normalized by the mean $\mathbf{m}$ and variance σ. The mean and variance are learned by estimating the sample-mean and sample-variance of the un-normalized feature vector $\mathbf{y}_n$ from the training set. The result is the normalized feature vector $\mathbf{x}_n$ presented to the classifier.

$$\mathbf{x}_n = \text{normalize}(\mathbf{y}_n) = \left[\frac{\mathbf{y}_{n,i} - \mathbf{m}_i}{\sigma_i}\right]_{i=1\ldots K} \qquad K = DL \tag{III.3}$$

We employ the Relevance Vector Machine classifier to take as input the normalized feature-vector and produce a classification among three classes. A left- and right- turn RVM are created for recognizing driver preparatory motions for left and right intersection turns. Each RVM is trained to produce a value of $+1$ for only patterns observed that lead up to a turn in the respective direction, and produce a value of -1 for all other patterns. A classification is found by computing the response of both classifiers and thresholding with a value τ according to the desired operating point. A classifier's operating point describes a mode of operation yielding a given true- and false-positive rate. If either response is greater than the chosen threshold τ, then a turn is predicted to occur. The direction is determined to be the one whose associated RVM classifier produced the greatest (most positive) response. If neither response is greater than τ, then no turn is predicted.

III.4.A Driver Intersection Turn Intent Inference Algorithm

The procedure is described below

$$\mathbf{y}'_n \overset{\text{step 2}}{\longrightarrow} \mathbf{y}_n \overset{\text{step 3}}{\longrightarrow} \mathbf{x}_n \overset{\text{step 4}}{\longrightarrow} \phi(\mathbf{x}_n) \overset{\text{step 5,6,7}}{\longrightarrow} \omega \tag{III.4}$$

Feature Vector Construction:

Step 1 Concatenate new observations $\{\mathbf{h}_n, \mathbf{l}_n, \mathbf{r}_n, \mathbf{c}_n\}$ into and remove the oldest observations from $\mathbf{Y}'_n \in \mathbb{R}^{D\times M}$, where each sample is time-aligned by linear interpolation, D is the dimension of the feature vector, and M is the sample window length.

Step 2 Let S be an integer multiple of M. Sub-sample the set of time-series $\mathbf{Y}'_n$ in time by S and vectorize the result to produce $\mathbf{y}_n$.

Step 3 Normalize the output from the previous step such that each element has a 0 mean and unit variance using the trained mean $\mathbf{m}$ and standard-deviation σ values learned from the training data to produce $\mathbf{x}_n$.

Kernel-RVM Classification:

Step 4 Project the feature vector onto the kernel space using the radial-basis-function kernel (RBF) with width parameter γ trained from data via cross-validation.

$$\boldsymbol{\phi}(\mathbf{x}) = \frac{\|\mathbf{x}_i - \mathbf{x}\|^2}{\gamma}$$

Step 5 Compute the dot-product of the resulting output kernel vector $\boldsymbol{\phi}(\mathbf{x}_n)$ with trained weight vector $\mathbf{w}_l, \mathbf{w}_r$ for the left and right kernel-RVM classifiers to produce responses

$$g_n^{(l)} = \mathbf{w}_l^\top \phi(\mathbf{x}_n)$$
$$g_n^{(r)} = \mathbf{w}_r^\top \phi(\mathbf{x}_n)$$

Step 6 If either RVM responses $g_n^{(l)}$ or $g_n^{(r)}$ exceeds the threshold τ, go to step 7. Else, decide no impending turn $\omega \leftarrow n$. (The value of τ is set at the value which produces the desired true and false-positive operating point.)

Step 7 If $g_n^{(l)} > g_n^{(r)}$, $\omega \leftarrow l$. Else, $\omega \leftarrow r$.

III.4.B Remarks on Choice of Classifier

The kernel-RVM classifier is a probabilistic cousin of the kernel-SVM or kernel Support Vector Machine. Together, they belong in the class of kernel-based approaches to the classification problem. Kernel approaches emphasize different aspects of the topology of the features in feature-space, making a nonlinearly separable feature space into a linearly separable one in a higher-dimensional kernel space, producing better classification performance.

The objectives of the two approaches are similar, and the attributes that distinguish one from the other are subtle yet significant. The objective is to learn the weights to the discriminant function

$$g(x) = \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}) \tag{III.5}$$

where $\boldsymbol{\phi}(\mathbf{x}) : \mathbb{R}^K \mapsto \mathbb{R}^N$ is the kernel function. Both SVM and RVM produce sparse solutions to $\mathbf{w}$, thereby requiring the classification algorithm to retain only a very small subset of the complete N vectors $\boldsymbol{\phi}(\mathbf{x})$. The remaining vectors are called *relevance vectors* for RVM and *support vectors* for SVM. The manner in which RVM arrives at the solution is fundamentally different and provides better properties as a result.

In the classification setting, the values of $g_{\text{SVM}}(\mathbf{x})$ represents the distance from the decision boundary in the feature space. However, $g_{\text{RVM}}(\mathbf{x})$ represent the probability that an observation belongs to one class or the other.

Also, RVM can be formulated as an M-class classifier by reformulating the likelihood function to consist of M-classes, unlike SVM which is intrinsically a 2-class classifier. In order to use SVM in an M-class classification setting, several SVM classifiers need to be trained and the results combined. Each SVM classifier would be trained using a one-versus-rest scheme, where each SVM will be trained to classify 1 of M classes as positive and the other (M-1) classes, negative. Classification then takes place by determining in which class did the feature vector (observation) lie deepest. This creates the drawback where observations that lie close to the decision boundary may not be modeled as well as they could be. In M-class RVM classification on the other hand, the M target class values and the associated observations can be utilized in the training process, optimizing the decision boundaries in one step–as opposed to training M-classifiers and combining the results afterward with none of the training of the SVM classifiers having knowledge of the *other* classes.

The one-versus-rest scheme is also the scheme used to deploy the kernel-RVM turn-intent classifier here. Reformulating the model to consider the three driving maneuvers in a multi-class classification problem is reserved for future work.

III.5 Experimental Evaluation

This section describes the evaluation of the driver intersection turn intent inference algorithm described in sec. III.4.

Positive examples are collected according to the specified sample window length M and the number of samples dt between the last sample in the window and the start of the intersection turn. Negative examples are collected outside of the positive examples every 2 seconds. The sensors produced data at 14Hz. Training was performed on 70% of the positive examples, and up to 500 of the negative

examples, all randomly selected from the positive and negative data-set. The training-set consists of 112 positive and 500 negative examples, while the validation-set consists of 47 positive and 1500 negative examples. These numbers were chosen to limit the training time to less than 10 minutes.

One of the performance measures used to evaluate the classifier is the receiver-operator-characteristics (ROC) curve. This is a plot that shows the true- and false-positive rate-pairs at which a classifier can operate given a particular threshold τ. The true- and false-positive rates are defined as

$$P(\text{true}|\text{true}) = \frac{\text{\# of True Examples Predicted as True}}{\text{\# of True Examples}} \tag{III.6}$$

$$P(\text{true}|\text{false}) = \frac{\text{\# of False Examples Predicted as True}}{\text{\# of False Examples}} \tag{III.7}$$

To span the entire range of operating points, the τ parameter is swept from the minimum to the maximum values ($g_n^{(l)}$ and $g_n^{(r)}$), and the true- and false-positive rates recorded.

There are a number of free parameters that influence the classification performance. We used the area underneath the ROC curve as the single value to gauge optimality when searching for the optimal set of parameters. These parameters are:

1. Decision time (dt)
2. Sub-sampling rate (S)
3. Window length (M)
4. Radial-Basis-Function Kernel width (γ)

The first three parameters pertain to the construction of the feature vector from raw sensed data. The width parameter affects the range of output values from the RBF. To determine the optimal values, a grid search is performed over these parameters.

In order to train the classifiers with a variety of parameter values in a reasonable amount of time, two parameter values are varied while the others are fixed. This accomplished two things: 1) Optimal values were verified for a decision time of 0 seconds, a classifier trained to make the best classification based on information just prior the vehicle entering the intersection at $t = 0$. 2) Performance was compared among 4 classifiers each given a different set of input cues.

The following lists how the variables are varied for each training run.

A. **Kernel-width (γ) vs. Window length (M).**

   Other parameters are fixed: Sub-sampling rate S$= 5$, and decision-time dt$= 0s$.

   $\gamma = \{3, 4, 5, 10, 15, 25, 40\}$

   $M = \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55\}$

B. **Kernel-width (γ) vs. Sub-sample Rate (S).**

   Other parameters are fixed: Window length M$= 20$, and decision-time dt$= 0s$.

   $\gamma = \{3, 4, 5, 10, 15, 25, 40\}$

   $S = \{2, 4, 5, 10, 20\}$

Table III.4: Driver Intent Classifier Cue Set

Cues	Set 1	Set 2	Set 3	Set 4
Head Orientation	○	○	○	×
Head Position	○	○	○	×
Left Hand Position	○	○	○	×
Right hand Position	○	○	○	×
Speed	○	○	×	○
Steering Angle	○	○	×	○
Steering Velocity	○	○	×	○
Torque	○	○	×	○
Throttle	○	○	×	○
Brake	○	○	×	○
Turn Signals	○	×	×	×

C. **Kernel-width (γ) vs. Decision-time (dt).**

Other parameters are fixed: Window length M= 20, and Sub-sampling rate S= 5.

$\gamma = \{3, 4, 5, 10, 15, 25, 40\}$

$dt = \{-28, -21, -14, -7, 0, 7, 14, 21, 28\}$

Data is collected at 14Hz, e.g. -7 represents -0.5 seconds from the moment the vehicle enters the intersection.

Training-runs A and B are for verifying the optimality of the $dt = 0$ classifiers. Training-run C is for comparison between classifiers trained with different input cues.

The above training-runs are performed with 4 different sets of input cues. Tab. III.4 summarizes the 4 sets of cues used. Sets 1 and 2 vary only by the inclusion of the turn-signals, the results from which will show the influence of turn-signal knowledge on turn intent inference. Sets 2, 3 and 4 are chosen to compare the influence of body pose and vehicle dynamics information by considering the classification performance given the two pieces of information individually and together.

The training results for input cue sets $1 \sim 4$ are shown in fig. III.6, III.7, III.8, and III.9. For each set of parameters, a classifier's ROC curve and area underneath the ROC curve are calculated. The area value is used to determine the shade and number on each square. The area under the ROC curve is the measure of optimality. For input cue sets 1, 2 and 4, the training runs show that the optimal parameter values for decision-time $dt = 0$ seconds are $\gamma = 5, M = 20, S = 5$ for the left turn, and $\gamma = 15, M = 20, S = 5$ for the right turn. Tab. III.5 summarizes the optimal parameter values. We arrive at $M = 20$ and $S = 5$ as optimal values for these runs by observing the maximum ROC areas in the results for training-runs A and B.

The grid search over 1) Kernel-width (γ), 2) Decision-time (dt) and 3) input cue sets, training-run C begins to paint a picture of the influence of different input cues on the task of predicting an intersection turn. This grid-search reveals the best values of γ to use in order to create a classifier that will

Table III.5: Optimal parameter values for intersection-turn classifiers with decision-time $dt = 0$ seconds. (Window length $M = 20$, Sub-sampling rate $S = 5$).

Input Cue Set	1		2		4	
γ	5	15	5	15	5	15
ROC Area	.93	.93	.94	.94	.96	.93

optimally make a decision at various decision-times with the given input cue set. Again, the decision-time dt is the time when the last sample of the window was collected relative to the start of the intersection turn maneuver. Tab. III.6 summarizes the results from training-run C over the 4 cue sets. Fig. III.10 depicts the same numbers in a graph.

The area under the ROC represents an ensemble performance result. Realistically, the classifier would be tuned to a single operating point in the ROC curve, such as shown in fig. III.11. The area thus represents the amount of flexibility a designer has for a particular classifier. The closer the area is to 1.0, the greater the flexibility.

Using the ROC Area as the performance measure, cue set 4 (vehicle dynamics only) provides the most flexibility for the left turn. Cue sets 1 and 2 compete with classifiers using cue set 4 for the right turn. One explanation for why cue sets 1 and 2 performed better for the right than for the left lies in the higher correlated patterns of head movements for the right turn than in the left turn, as can be seen in fig. III.1. The cue set that operates consistently poorer is cue set 3 (body pose only). This was not too surprising considering the noisy nature of the data, which could account for the diminished performance when considering vehicle dynamics together with body pose in cue set 1 and 2. Over-training may have occurred and thus degraded the final classification performance.

When the ROC areas are compared between cue sets 1 and 2 (with and without turn-signals), we can clearly see that the turn-signal cue does not uniformly improve the classifier performance. In fact, for most decision times, turn signals can be considered redundant information for the pattern classifier. There is an exception at $dt = 1s$ and $dt = 1.5s$ for the left turn and $dt = 1.5s$ and $dt = 2s$ for the right turn. With the classifier presented, turn-signals information is does not help, but can help if a different pattern recognition model was used. For example, the turn signals can dictate the use of 1 of 2 classifiers, one tuned optimally classify turn intent in the presence of turn signal activation, and the other, without.

The ROC area is a convenient measure for comparing classifiers with a scalar number, but does not immediately convey its performance felt by the driver. For this, the actual ROC curves are needed. Fig. III.11 shows the ROC curves of the optimal classifiers for the 4 input cues trained for a decision time of $dt = 0$s. The ROC curve is the true-positive rate (y-axis) plotted as a function of the false-positive rate (x-axis in log-scale). At 10% false-positive, the classifier utilizing cue set 4 (vehicle dynamics information only) can achieve 80% and 86% true-positive rate for the left and right turns respectively. In a system that operates at 10 Hz, this implies every 1 second, 1 instance in time is mistaken to be positive (impending

Table III.6: Area under the ROC curve for classifiers trained for different decision-times (dt). A negative decision time indicates the amount of offset from the time when the vehicle enters the intersection. (Window length $M = 20$, Sub-sampling rate $S = 5$, and optimal kernel-widths γ)

(a) Left Turns

	Cue Set			
dt [sec]	1	2	3	4
-2.0	0.8245	0.8100	0.7616	**0.8730**
-1.5	0.8003	0.8642	0.7610	**0.8931**
-1.0	0.8661	0.8800	0.7734	**0.9210**
-0.5	0.8737	0.9168	0.8877	**0.9450**
+0.0	0.9282	0.9393	0.8618	**0.9554**
+0.5	0.9486	0.9560	0.9396	**0.9696**
+1.0	0.9565	0.9492	0.9069	**0.9644**
+1.5	0.9599	0.9562	0.9246	**0.9684**
+2.0	0.9524	**0.9595**	0.9547	0.9594

(b) Right Turns

	Cue Set			
dt [sec]	1	2	3	4
-2.0	0.7803	0.7986	0.6753	**0.8729**
-1.5	0.8198	0.7970	0.7387	**0.9044**
-1.0	0.8745	0.9108	0.6216	**0.9370**
-0.5	0.8961	0.9224	0.8451	**0.9230**
+0.0	0.9265	**0.9395**	0.8663	0.9297
+0.5	0.9391	**0.9456**	0.9230	0.9400
+1.0	0.9476	0.9498	0.9224	**0.9513**
+1.5	**0.9606**	0.9542	0.9304	0.9466
+2.0	**0.9531**	0.9497	0.9462	0.9218

turn) when in fact negative. With a more stringent 2% false-positive or 1 error in every 5 seconds, the true-positive rate dips to 50% and 60%. At 1% (a false-alarm every 10 seconds), the performance is 20% and 40% for the left and right turn respectively. These rates may look poor, but one must keep in mind that only 2 seconds worth of vehicle dynamics information is used, and only information prior to the moment the vehicle enters the intersection ($t = 0s$) is used to make each determination.

When we examine the ROC curves, the differences between classifiers and their impact on real-world reliability become clear. Cue set 4 classifiers can provide markedly better true-positive rates for a given false-positive rate for the left-turn. The classifiers of all cue sets performed about the same for the right-turn case.

To understand how the classifiers perform around the time of the intersection-turn event, the classifier responses surrounding the start of the intersection-turn are overlaid on top of one another. The statistics of the responses at each point in time are generated. At each time t the height of the response where 10% of the other responses lay underneath is found. The process is repeated for 20%, 30%, 40% and so on. A line then connects the points with the same percentages, resulting in percentile lines. We refer to this as the time-aligned statistical response plot. Only classifier responses along intersection-turn examples in the validation data-set are used to generate this plot. The time-aligned statistical response plots are shown in fig. III.12 and III.13 for the classifiers utilizing 1 of the 4 cue sets. Each line represents a percentile line. The black shaded area represents the range of response values at that time instant. The upper and lower boundaries of the shaded area represent the maximum and minimum response values at that time instant, i.e. 0% and 100% percentile lines.

The advantages of this visualization are that it enables the reader to see at a glance what proportion of turns were correctly and incorrectly classified, and how early the proportion of turns were classified. Take fig. III.12(d) for example. If one used a threshold $\tau = -0.5$ in this case, one can see

that 90% of the turns at $t = 0$ are above -0.5. Of these intersection-turns, 10% of the turns would have exceeded the threshold at $t = -2$. Interpreted another way, 10% of all turns would be recognized 2 seconds before the vehicle entered the intersection. A full 80% would have been recognized 0.5s before the vehicle entered the intersection.

Because only a window of 14 seconds surrounding the intersection-turn is depicted here, ROC curves are still needed to gauge the recognition performance for entire data-sequences. Valid neighboring intersection-turns would appear in these plots as well. So the time-aligned statistical response plot is useful for a limited number of seconds around the turn events.

III.6 Discussion and Concluding Remarks

In this chapter, we introduced a characterization of a type of intersection turn maneuver called the "slow" turn, which represented a large 93.7% of all turns in our data-set. This type of intersection turn possessed relatively consistent trends in terms of body pose and vehicle dynamics moments before and during the turn maneuver. We devised an integrated system to infer the driver's intent to perform an intersection turn maneuver, by presenting a kernel-RVM classifier with data including driver head and hand pose information as well as 8 vehicle dynamics parameters (speed, throttle, torque, brake activation, left/right turn signals, steering angle and steering angular velocity). Experimental evaluations show that the system was capable of inferring a majority portion of the intersection turns *before* the vehicle enters the intersection ($dt = 0$): up to 100% correct detection rate at 20% or more false alarm rate, or better than 80% correct, at 7% false alarm.

We showed that turn-signals do not improve the performance of the intersection turn intent classifier. In the interest of recognizing turn-intent in the absence of turn-signals, we conclude omitting those signals is beneficial at least for the presented classifier. This lifts any requirement that the training data-set contain a mixture of turn-instances with and without turn signals.

We showed that actual body head and hand pose information, as it was utilized in this turn-intent classifier, does not add value to the final correct classification performance. Other cues derived from actions of the driver were better suited to capturing the movements of the driver that indicate turn-intent. These other cues are throttle position, steering angle, steering angular velocity, and brake activation. In terms of the most appropriate cues in the 2 seconds before the turn, the historical trajectories of these cues yield the best classification (driver turn-intent inference) performance.

Lastly, the proposed system represents a case-study of an improving framework for the development and evaluation of driver-intent inference systems. Beyond the ROC curve, which disregards time in the measurement of performance, we introduced a new visualization of the soft-response of the classifier that better shows the behavior of the classifier, which overlays all the responses of the classifier over time,

aligned to the start of the turn maneuver. With this measure, we are able to gauge proportion of turns whose response rose above the threshold τ before the intended decision time dt.

III.7 Future Work

Many aspects of the work presented in this chapter can be extended. Several directions are presented.

III.7.A Temporal cues.

As noted above in sec. III.5, there exists trends in the body pose sequences that may not be adequately modeled by the kernel-RVM pattern classifier. Specifically, the left hand is observed to begin its movement in the y-direction as early as -2s and as late as 1s relative to the start of the intersection turn. The RBF kernel-RVM classifier cannot be trained to observe such cues reliably without additional assumptions (or more training data) because the classifier depends on when the movement started and ended and if that pattern existed in the training set. This makes the positive examples less distinct from other negative patterns, increasing the intra-class variance. Future work includes appropriate feature pre-processing or changes in the model that can account for this.

The HMM may be more appropriate here. With that said, there is value in utilizing the definition of the intersection-turn to better describe the proportion of predicted turns that actually matter, or the ones that were predicted precisely before the vehicle entered the intersection.

III.7.B Automatic Feature Selection.

One of the unique aspects of kernel-RVM is its ability to automatically determine the relevance of the basis vectors. The radial-basis-function was used in this study. However, if a different basis-function were used, one could potentially frame the training of the kernel-RVM as simultaneously determining which input cues (head, hand, speed, throttle, etc.) are the most relevant to the recognition of driver intent [38].

The text of Chapter III, in part, is a reprint of the material as it appears in: Shinko Y. Cheng, Mohan M. Trivedi, “Turn-Intent Analysis Using Body Pose for Intelligent Driver Assistance”, IEEE Pervasive Computing, vol. 5, number 4, pages 28-37, Oct-Dec 2006. I was the primary researcher of the cited material and the co-author listed in this publication directed and supervised the research which forms the basis of this chapter.

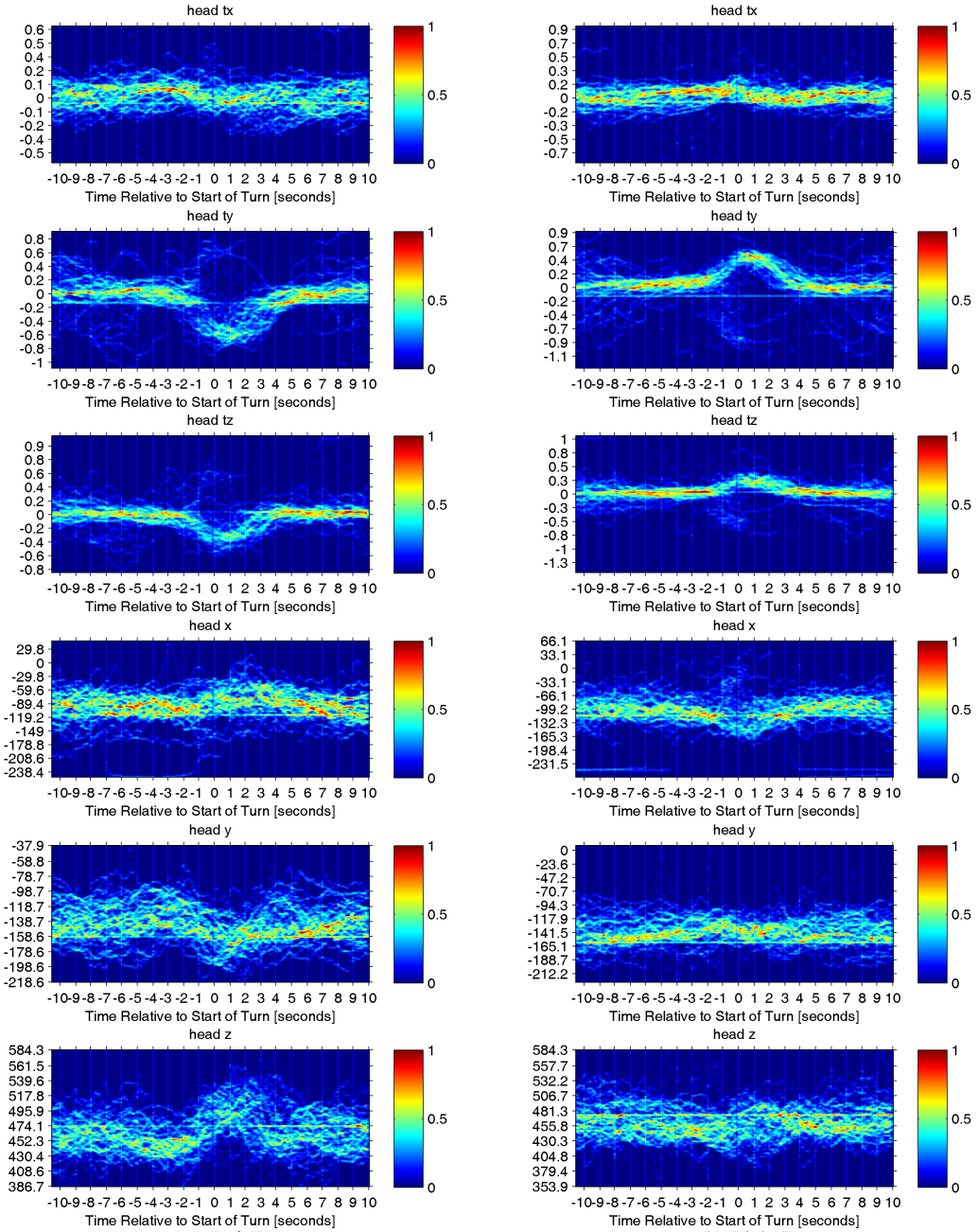
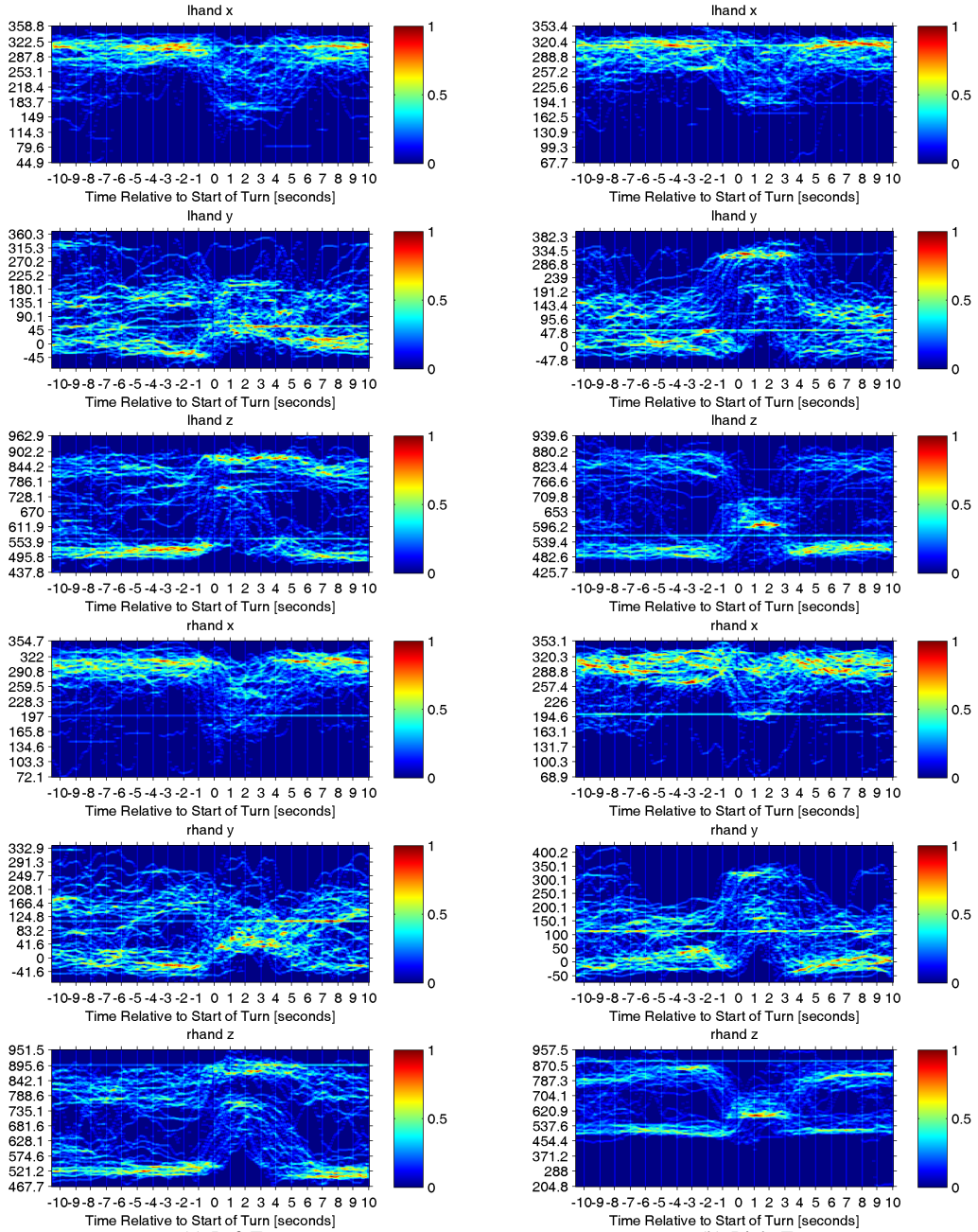
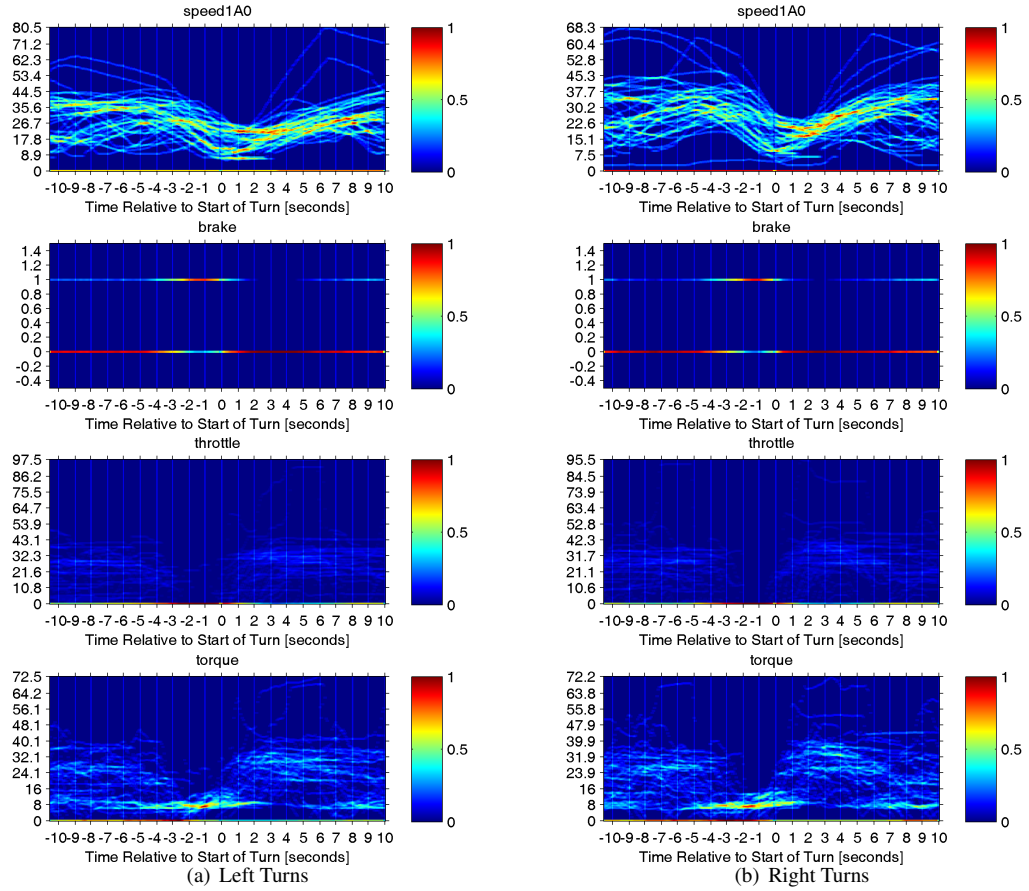


Figure III.1: A histogram of head-pose time-sequence. Each instance is time-aligned to the start of the intersection turn $t = 0$. The plots show orientation $(\theta_x, \theta_y, \theta_z)$ and position sequence (x, y, z) of the head. Warmer colors indicate more instances following that specific sequence of values. A pattern is perceivable during and about 1 second before the start of the intersection turn.



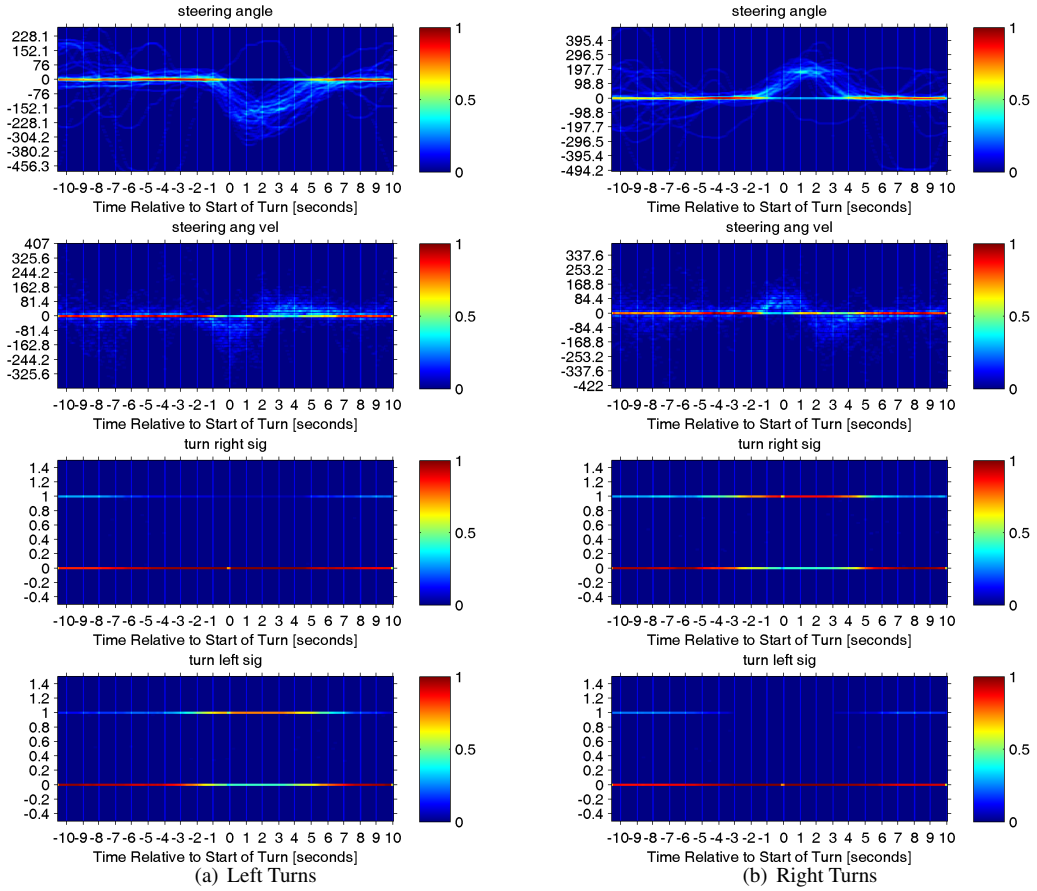
(a) Left Turns (b) Right Turns

Figure III.2: A 2-D histogram of hand position time-sequences. Each instance is time-aligned to the start of the intersection turn $t = 0$. The plots show position sequence (x, y, z) of the left and right hands. Warmer colors indicate more instances following that specific sequence of values. A pattern is perceivable during and about 1 second before the start of the intersection turn.



(a) Left Turns (b) Right Turns

Figure III.3: A 2-D histogram of vehicle speed, steering angle, steering angular velocity and torque time-sequences. Each instance is time-aligned to the start of the intersection turn ($t_o = 0$). Warmer colors indicate more instances following that specific sequence of values. A pattern is perceivable during and about 1 second before the start of the intersection turn.



(a) Left Turns (b) Right Turns

Figure III.4: A 2-D histogram of throttle, brake activation, left turn signal, and right turn signal time-sequences. Each instance is time-aligned to the start of the intersection turn ($t_o = 0$). Warmer colors indicate more instances following that specific sequence of values. A pattern is perceivable during and about 1 second before the start of the intersection turn.

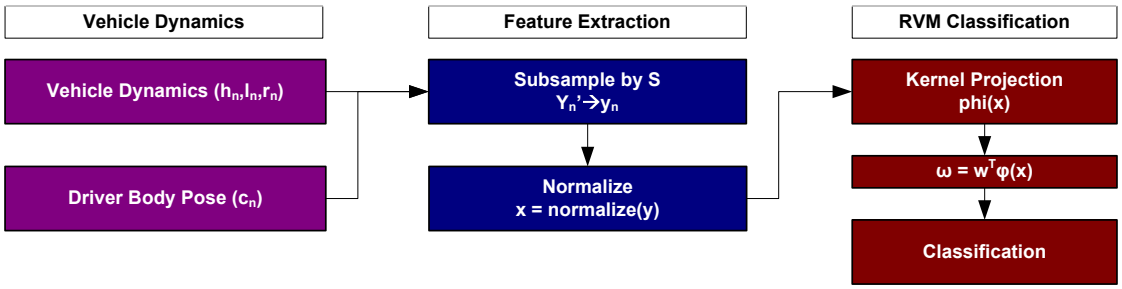
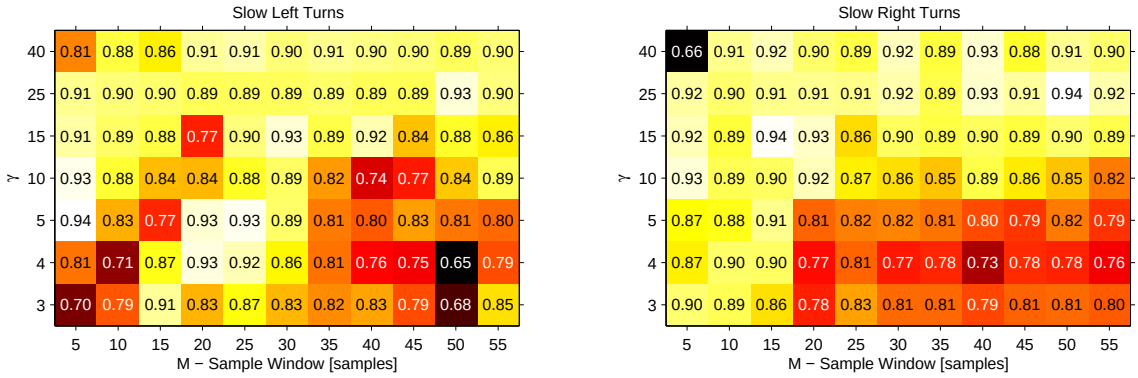
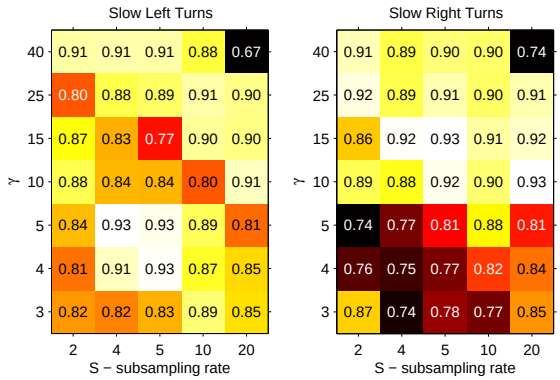


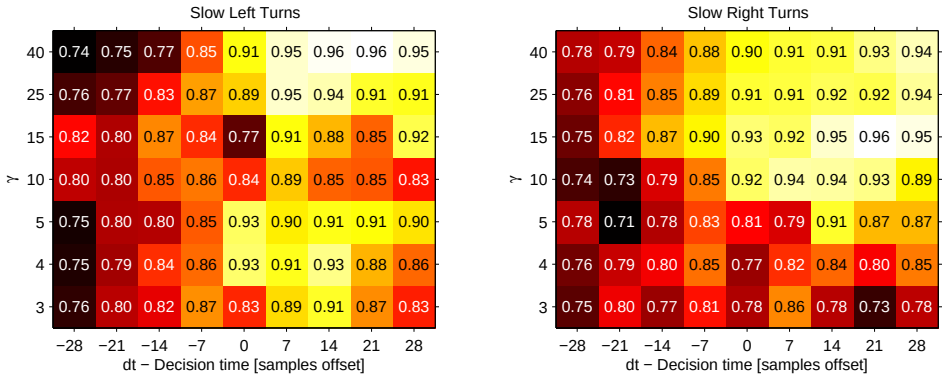
Figure III.5: Process flow of RVM based Driver Intent Recognition.



(a) Kernel-width (γ) vs. Window Length (M), (S=5, dt=0s)

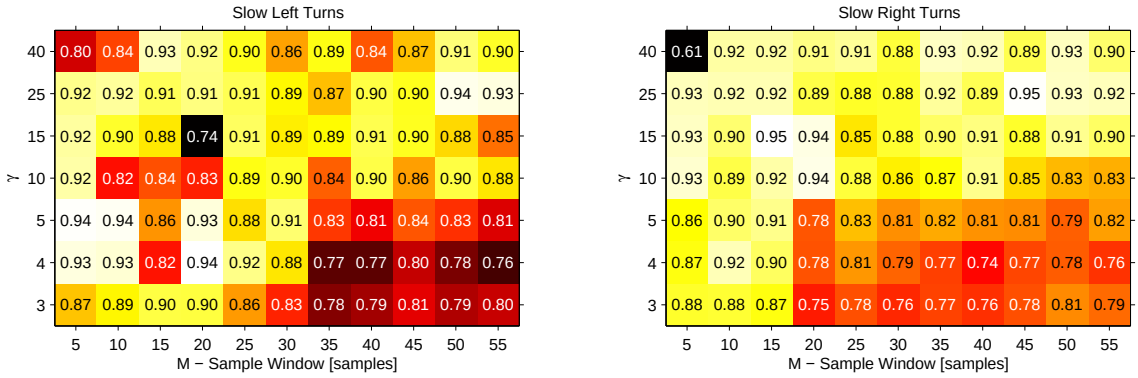


(b) Kernel-width (γ) vs. Sub-sample Rate (S), (M=20, dt=0s)

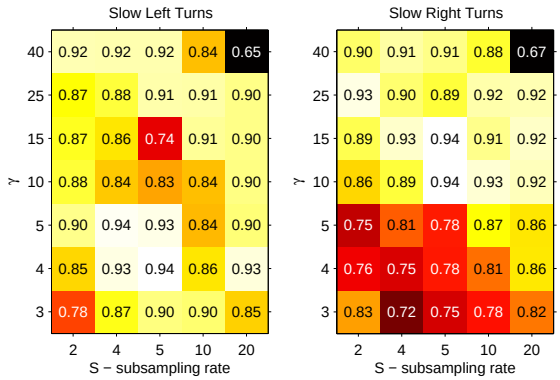


(c) Kernel-width (γ) vs. Decision time (dt), (M=20, S=5)

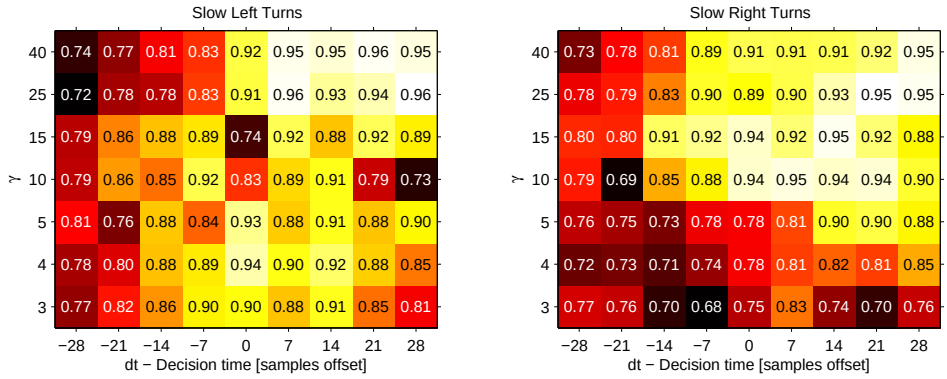
Figure III.6: Grid search results for Cue Set 1. Box numbers represent area under the ROC curve.



(a) Kernel-width (γ) vs. Window Length (M), (S=5, dt=0s)

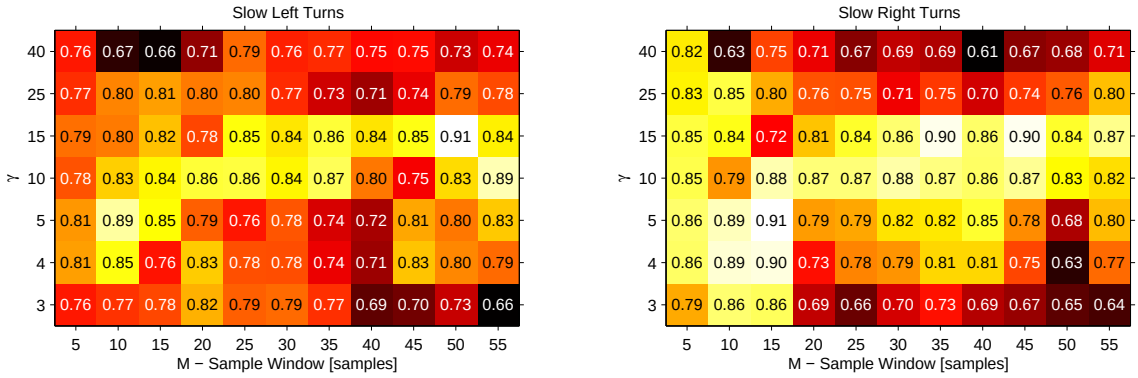


(b) Kernel-width (γ) vs. Sub-sample Rate (S), (M=20, dt=0s)

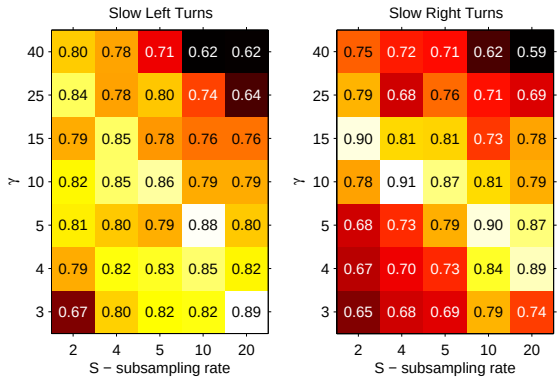


(c) Kernel-width (γ) vs. Decision time (dt), (M=20, S=5)

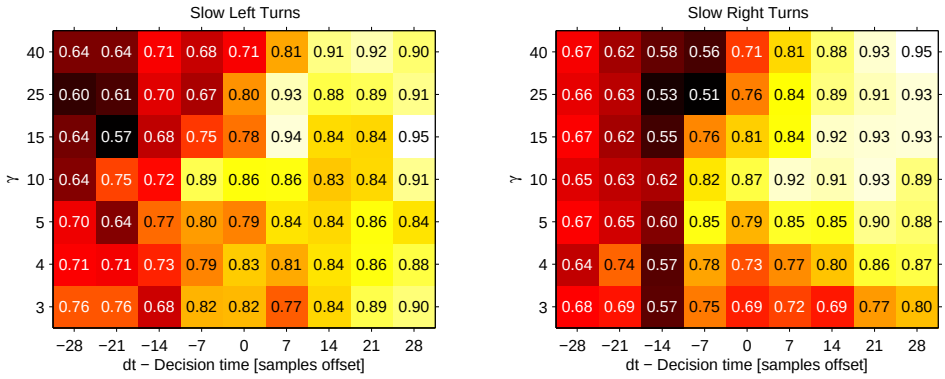
Figure III.7: Grid search results for Cue Set 2. Box numbers represent area under the ROC curve.



(a) Kernel-width (γ) vs. Window Length (M), (S=5, dt=0s)

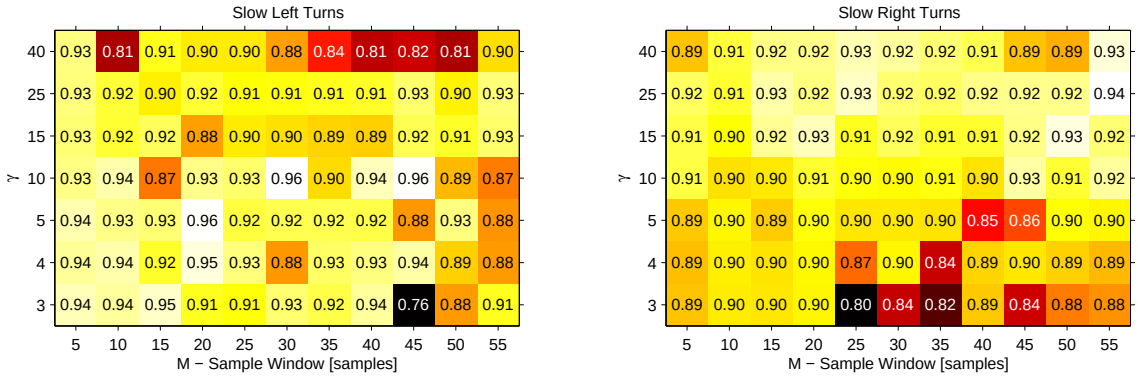


(b) Kernel-width (γ) vs. Sub-sample Rate (S), (M=20, dt=0s)

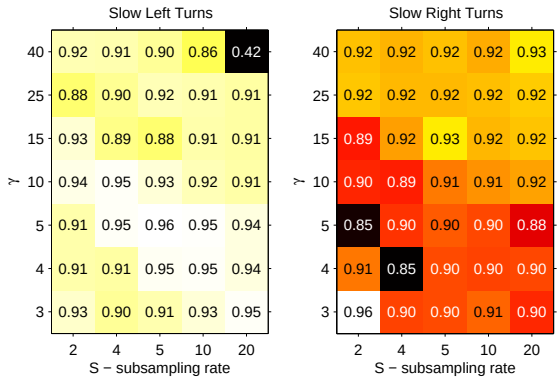


(c) Kernel-width (γ) vs. Decision time (dt), (M=20, S=5)

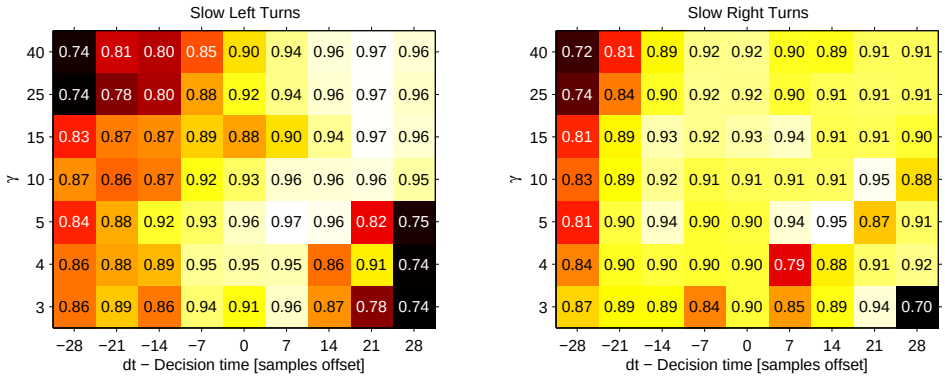
Figure III.8: Grid search results for Cue Set 3. Box numbers represent area under the ROC curve.



(a) Kernel-width (γ) vs. Window Length (M), (S=5, dt=0s)



(b) Kernel-width (γ) vs. Sub-sample Rate (S), (M=20, dt=0s)



(c) Kernel-width (γ) vs. Decision time (dt), (M=20, S=5)

Figure III.9: Grid search results for Cue Set 4. Box numbers represent area under the ROC curve.

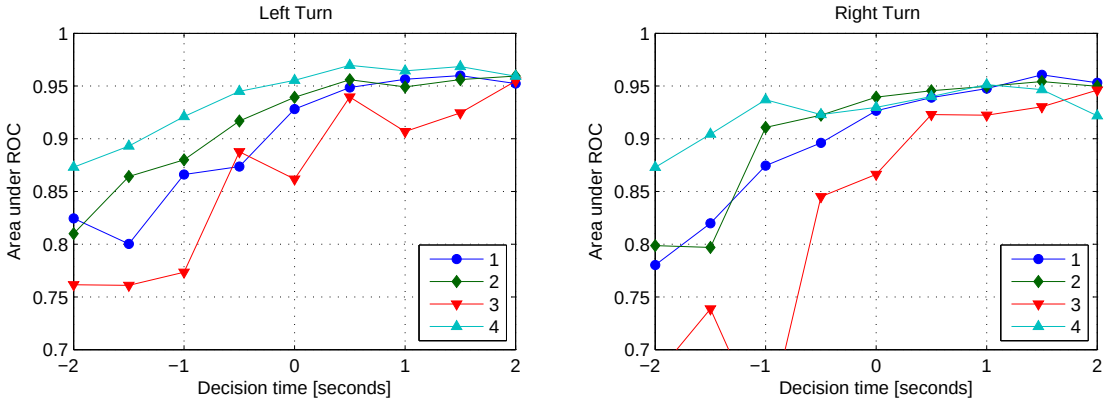
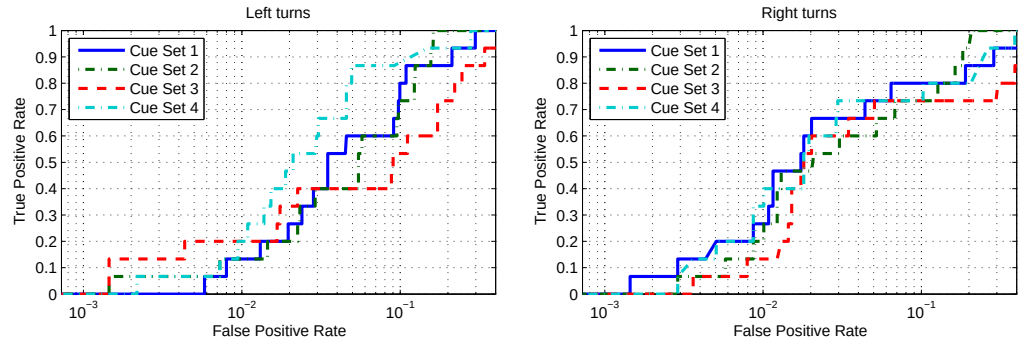


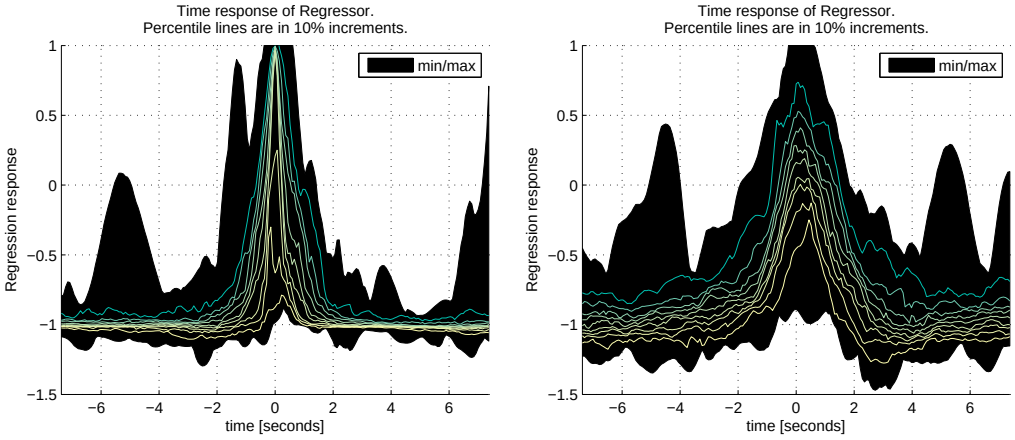
Figure III.10: Area under the ROC plot against Decision time for classifiers trained with window length $M = 20$, sub-sampling rate $S = 5$ and optimal kernel-widths γ as determined from the grid-search in training-run C.



(a) Left turn

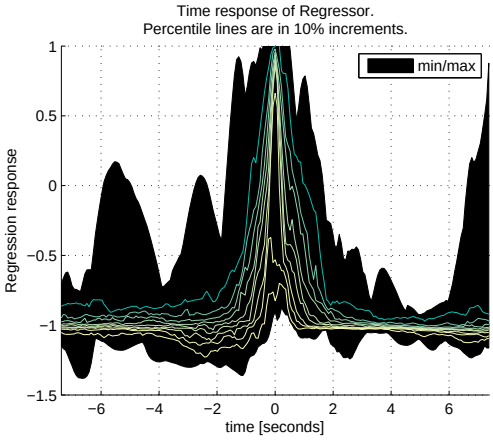
(b) Right turn

Figure III.11: Receiver Operator Characteristics curves for classifiers trained for various decision times. Each classifier is trained with an optimal kernel-width γ. Window length $M = 20$ and Sub-sampling rate ($S = 5$).

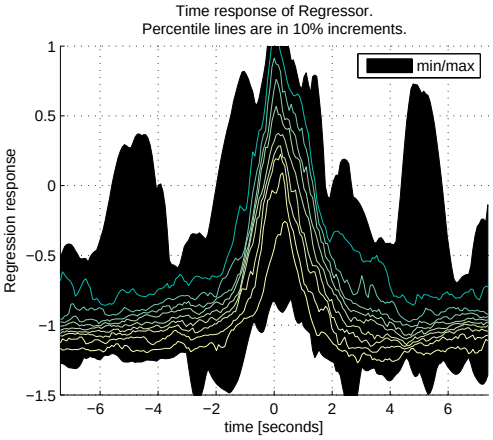


(a) Left turn: Cue set 1

(b) Right turn: Cue set 1

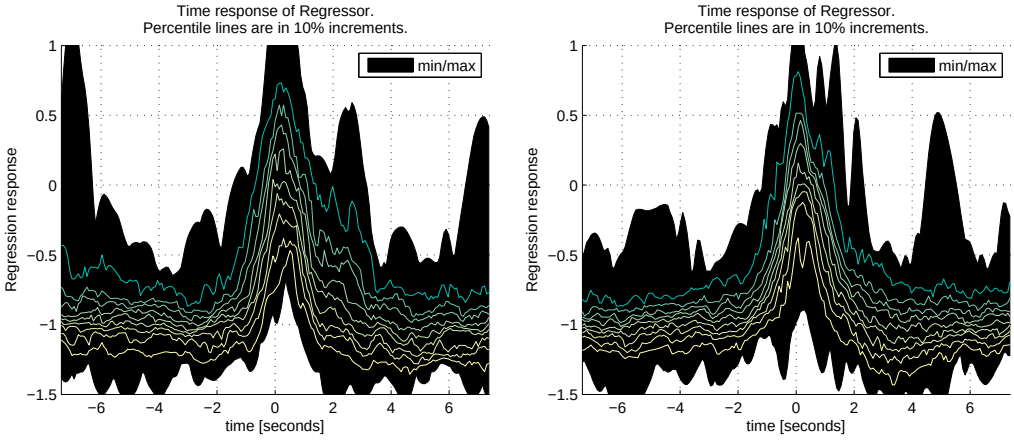


(c) Left turn: Cue set 2



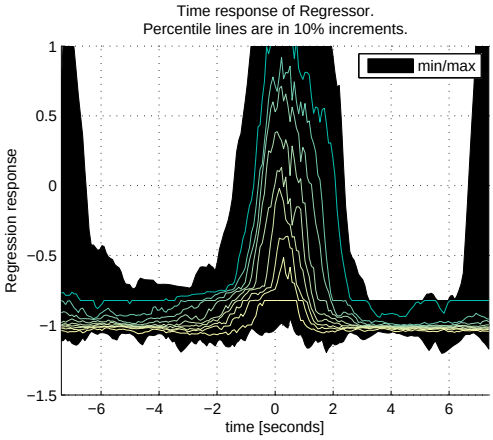
(d) Right turn: Cue set 2

Figure III.12: Time response of kernel-RVM intersection turn classifier using Cue Sets 1 and 2.

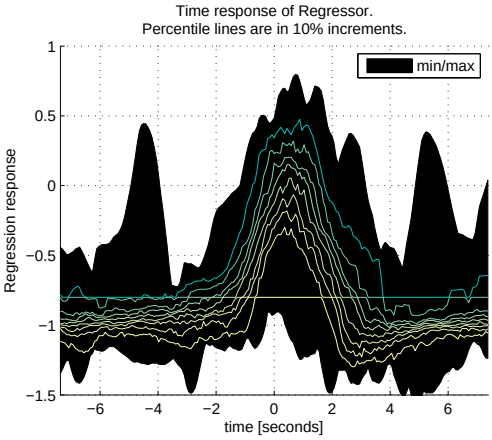


(a) Left turn: Cue set 3

(b) Right turn: Cue set 3



(c) Left turn: Cue set 4



(d) Right turn: Cue set 4

Figure III.13: Time response of kernel-RVM intersection turn classifier using Cue Sets 3 and 4.

IV

In-Vehicle Driver Hands Tracking in Infrared Imagery

The in-vehicle environment presents special requirements on driver hand tracking. In this chapter we present a computer vision algorithm that addresses these requirements for tracking the driver's hands while manipulating the steering and front console controls and present an application in steering wheel grasp analysis.

IV.1 Introduction

The Driver Hands Tracking system is intended to fill the requirement of body part position tracking in the vehicle for improving the safety of the vehicle. The premise of this thesis has been that there exists useful gestural information in the driver's pose that the vehicle system can make use of to improve driving safety. The vehicle system would use this information to determine the attentive state of the driver in critical situations. The challenge is to develop techniques to extract this pose and gesture information of the driver while in the vehicle and when the vehicle is in motion.

The challenge associated with object tracking in a moving vehicle is the varying nature of illumination. These variations are due to the sun and other sources of light illuminating the interior in unpredictable ways. The illumination enters from various incident angles resulting from the continuous movement of the vehicle, consists of various intensities modulated by atmospheric effects in the environment like cloud cover, contains abrupt changes in intensity due to obstructions by objects in the environment and the vehicle itself, and causes sharp contrasts in the scene during the day that are difficult to capture for many digital imaging sensors of limited dynamic range. All of these variations can be sensed by the imaging devices together with the intended objects, and are considered the "noise" affecting

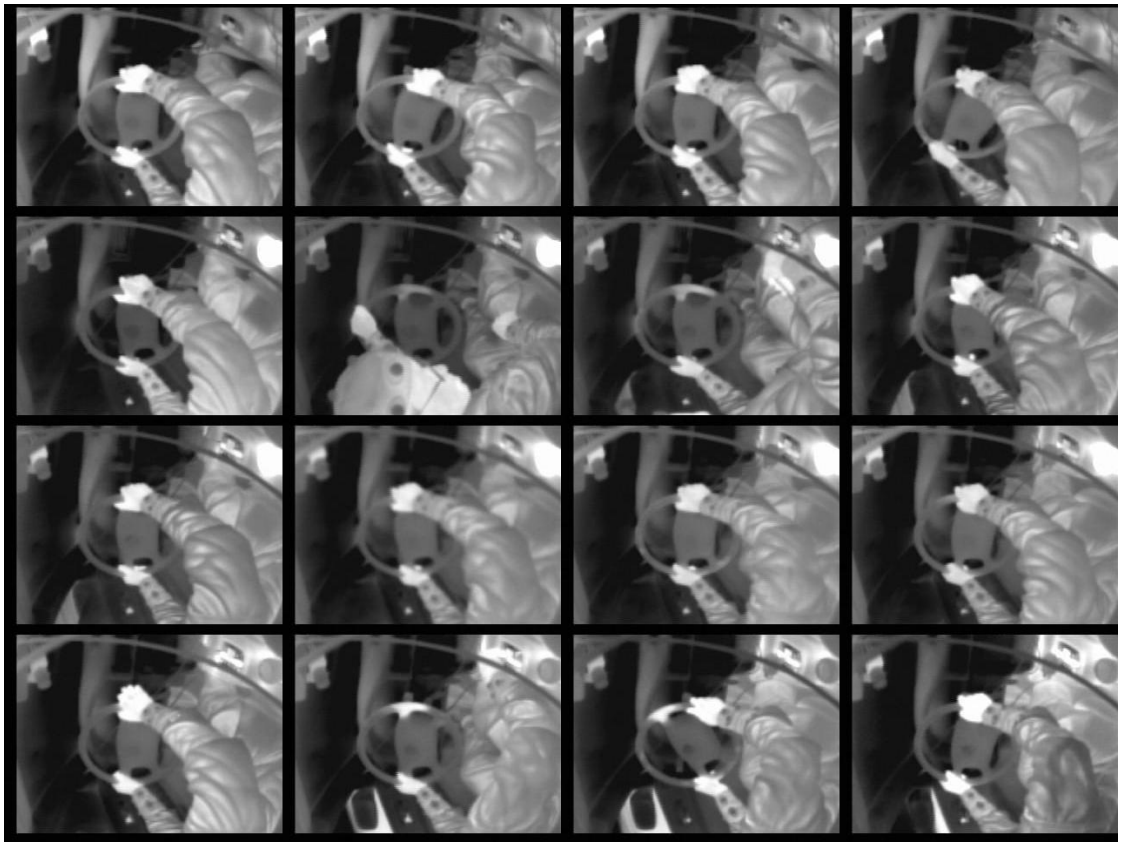


Figure IV.1: Infrared images taken over 90 minutes of driving. Note that there is very little variation in the gray levels of the subject and scene, besides where there is movement of objects.

the appearance of the observed objects in the vehicle. A requirement of high importance is to overcome these problems by devising algorithms that perform well in the presence of this noise.

We propose a system that utilizes the long-wavelength infrared (LWIR) cameras to detect the movement of the driver's hands. The heat sensing attribute of the thermal infrared camera is especially appropriate for use inside a vehicle where visible illumination is constantly changing. LWIR images do not exhibit problems associated with visible illumination changes since the camera senses emitted thermal-band electromagnetic radiation ($25 - 350\mu m$) from object surfaces. A change in temperature indeed results in a change of the level of thermal radiation. However this was observed to occur much slower in comparison to visible illumination changes. Visible illumination often changes faster than the frame rate of the camera. Fig. IV.1 illustrates the strength of LWIR imaging. The series of images shows the stability of intensity values of the subject's hands and scene for over 90 minutes of driving during an afternoon under the sun. This has simplifying implications on the algorithms, and is used to extract hand position information.

IV.2 Hand Detection and Tracking

IV.2.A Hand Detection

We define hand position in the vehicle to be the position of the hands in the LWIR image. The image locations of the hands are first extracted using the object detector proposed by Viola-Jones [39]. This method performs an exhaustive search of an image by testing every point of an image of several sizes. The speed is acquired by two things: 1) using Haar-wavelet like feature descriptors to describe each image patch and 2) using a cascade of boosted classifiers to classify each candidate image patch. The features can be very efficiently computed from the image when the image is first transformed into an *integral image.* This step reduces the required computations for extracting the feature description of the image patch to at most 9 memory access operations from at least 100 for a small 10x10 image patch [40]. Classification of each candidate image patch is performed using a cascade of boosted ensemble STUMP classifiers. The result of the cascade is a speed up gained by quickly throwing out negative patches at the beginning of the cascade, and concentrating the computations on more difficult patches with additional classification stages of the cascade. Each ensemble STUMP classifier furthermore only uses a subset of all the possible features that can be calculated from an image patch reducing the computation of each stage of the cascade.

Fig. IV.2 shows examples of the 20x20 pixel positive sample thermal hand images used for training and testing. These were extracted by hand from video captured from the LISA-P experimental test-bed vehicle (For details of the data capture test-bed, see app. B). Negative examples are randomly chosen from the same video sequence everywhere except the marked hand locations. The most salient features chosen in the first 3 stages of the cascade classifier are shown in fig. IV.3.

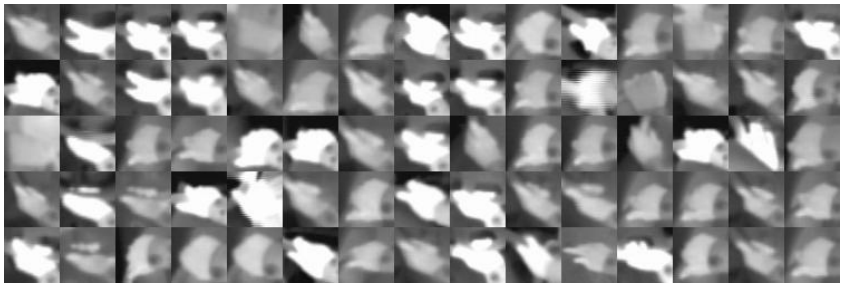


Figure IV.2: Positive example LWIR images of hands of drivers. A total of 2153 examples were used.

Often, exactly two candidate image regions are detected as hands, but occasionally more than two, one or no hands are detected when there should be two. To discern which of these multiple detections are truly hands and which among those are the left and right hands, we utilize a combination of kinetic information of the hands and their appearance. The detected hand candidates are tracked using a constant

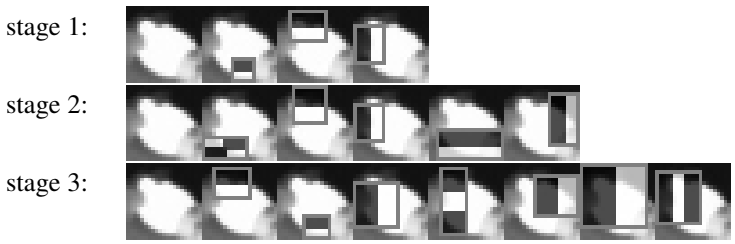


Figure IV.3: Features used in the first three stages of the classifier cascade for hand detection in LWIR images.

velocity Kalman Filter with Probabilistic Data Association (PDA) Filter. Multiple hand targets are maintained, as will be explained in the next section, by producing as many tracks as necessary to accommodate all "unclaimed" measurements. The most likely targets are classified as a left or right hand by examining (1) the prior probability that a left or right hand is present on their respective sides of the steering wheel, (2) the similarity of the appearance of the target with the appearance of the last recognized left or right hand target, and (3) the longevity or confidence of the track.

The detection produces the position (x, y) and bounding-box width w of the image region detected as a hand (both left and right). This is taken as the measurement for the Kalman Tracking, $\mathbf{z}_{t,m} = (x, y, w)^\top$ where $m \in \mathcal{M}_t$ is one in a set of $\mathcal{M}_t$ measurements at time t. The state vector of the Kalman tracker for each track n is defined by $\mathbf{x}_{t,n} = (x, y, v_x, v_y, w)^\top$ where (x, y) is the position, w is the size, and (v_x, v_y) is the velocity of the target hand in the image.

IV.2.B Multi-Target Tracking and Left/Right Classification

Fundamental to the target tracking algorithm is data association, which associates candidate measurements with tracked targets. There are a number of ways to establish the correspondence between candidate measurements and targets, e.g. Global Nearest Neighbor, Probabilistic Data Association, and Multiple Hypothesis Testing methods [41, 42]. Because of the sparse yet spurious errors in detection and momentary missed detections of the correct image regions, we adopt the probabilistic data association method. This method collects the measurements within a specified gate, or proximity to the predicted target location, which is modulated by the estimated measurement error covariance matrix, and compares the proximity of these measurements to the predicted target location in the presence of Poisson noise. The resulting track score represents the likelihood that the measurement belongs to the track, and the measurement's likelihood to be part of the spurious background detections. These likelihoods represent the confidence of each measurement, and are the weights applied to each measurement in calculating the innovation in the correction step.

Measurements that are not associated to any target during the gating process are considered

new potential targets. All potential and valid targets accumulate a track confidence score E. This score represents the proportion of times the track had a valid measurement with which to correct the estimate. This is calculated depending on whether or not a valid measurement is present to correct the estimate. In other words,

$$E_t = \begin{cases} \alpha_E E_{t-1} + (1-\alpha_E)\cdot 1 & \exists \text{ valid measurement} \\ \alpha_E E_{t-1} + (1-\alpha_E)\cdot 0 & \nexists \text{ valid measurement} \end{cases}$$

and α_E is the forgetting factor. Upon exceeding an empirically obtained threshold τ_E, a potential target is considered a valid target. Likewise, a target may lose track when the target does not have a valid measurement at the point when E dips below the threshold.

All potential and valid targets also maintain an adaptive appearance model $T_i \in \mathbb{R}^{M\times M}$ of the image of the hand which is updated by interpolating the detected image patch to the preset size $M \times M$ and incorporated into T_i using another first order autoregressive model (forgetting factor α_T). Similarly, appearance models are used to describe the appearance of the left and right hands, U_L and U_R. These are updated based on the appearance of the classified left and right hand targets, with a forgetting factor α_U. The appearance models U_L, and U_R are initialized to zero.

All potential and valid targets also accumulate a left hand and right hand likelihood measure. This measure consists of two quantities: (1) a target's proximity to the left or right hand's usual position in the driver's area, and (2) the similarity in appearance of the target's appearance model with the stored left and right hand appearances U_L and U_R. The first quantity is the proximity of the target to the likely locations either the left or right hand in the image which is *a priori* known. These locations are over the left and right side of steering wheel as illustrated in fig. IV.4. This first quantity is modeled as a bi-variate Gaussian probability, with log-likelihood values for the left and right hand given by $l_{t,p} = \log P(\mathbf{x}_t|\mu_L, \Sigma_L)$ and $r_{t,p} = \log P(\mathbf{x}_t|\mu_R, \Sigma_R)$, where$\mu_L, \mu_R \in \mathbb{R}^2$. The normalized sum of squared difference is used as a measure of similarity between the appearance model of the target T_i and the left U_L and right U_R hands. Together, the two quantities form a likelihood that the target is a left or a right hand

$$\begin{aligned} l_t &= \log P(\mathbf{x}_t|\, \mu_L, \Sigma_L)\cdot(1 - NSSD_L) \\ r_t &= \log P(\mathbf{x}_t|\, \mu_R, \Sigma_R)\cdot(1 - NSSD_R) \end{aligned}$$

where $NNSD_h = \|T_i - U_h\|^2/M^2$ and $M \times M$ are the dimensions of the appearance model. The amounts are accumulated with a forgetting factor α_s according to the relation

$$L_t = \alpha_s L_{t-1} + (1-\alpha_s) l_t \qquad R_t = \alpha_s R_{t-1} + (1-\alpha_s) r_t$$

A target with a large left hand score relative to the right-hand score indicates that the target has hovered over the likely left hand position longer than in the likely right hand position in the image. A higher value

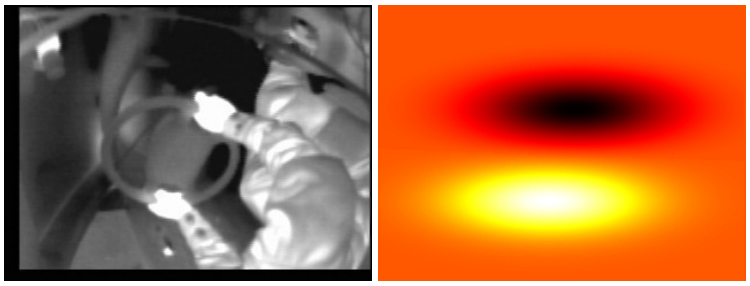


Figure IV.4: Hand location prior probabilities are bi-variate Gaussian density functions centered at the mean location of the hands in the image with a fixed variance. The two prior probabilities are illustrated as white and black regions of the image. The left and right hands are most likely in the white and black regions of the image

of the left hand score relative to all other target left hand scores represents a higher amount of confidence and freshness of the target being and having been the left hand. Finally, among the valid targets ($E \geq \tau_E$), the targets with the highest left hand score and right hand score are classified as the left and right hand, respectively.

IV.3 Experimental Evaluation

Hand detection and tracking results are shown in fig. IV.5. At each time step, appearance models are maintained, and updated as image patches are accumulated from the left and right hand recognition. The row of image patches beneath the tracking illustration are the left hand, right hand, and the various target appearance models. Since all detections that are not in the gate of the other targets cause a new track to be formed, outlier targets are also tracked to disambiguate true targets with false ones. These outlier tracks tend to come and go while the true targets remain consistently tracked throughout the sequence, punctuated with moments of loss of track.

At the start of the algorithm, the left and right hand appearance model is initialized to all zeroes. This places more importance on the proximity and duration a target lingers in the high likelihood left and right hand positions (in the image) rather than the stored appearance of the left and right hand in deciding which target is the left or right hand. Then, as the appearance model is slowly updated at each time step following the target recognition, the appearance model U_L and U_R play an increasingly influential role in deciding which targets are left and right hands. This then allows the tracking algorithm to decide targets as hands even in the rare moments when the hand targets depart from the hand's assumed usual location.

To gain an understanding of the observable pattern of hand movement, we collect segments of hand position data surrounding the intersection turn, time-align them at the moment the driver enters the intersection, and generate a 2-D histogram of those segments. A total of 12 such histograms were created

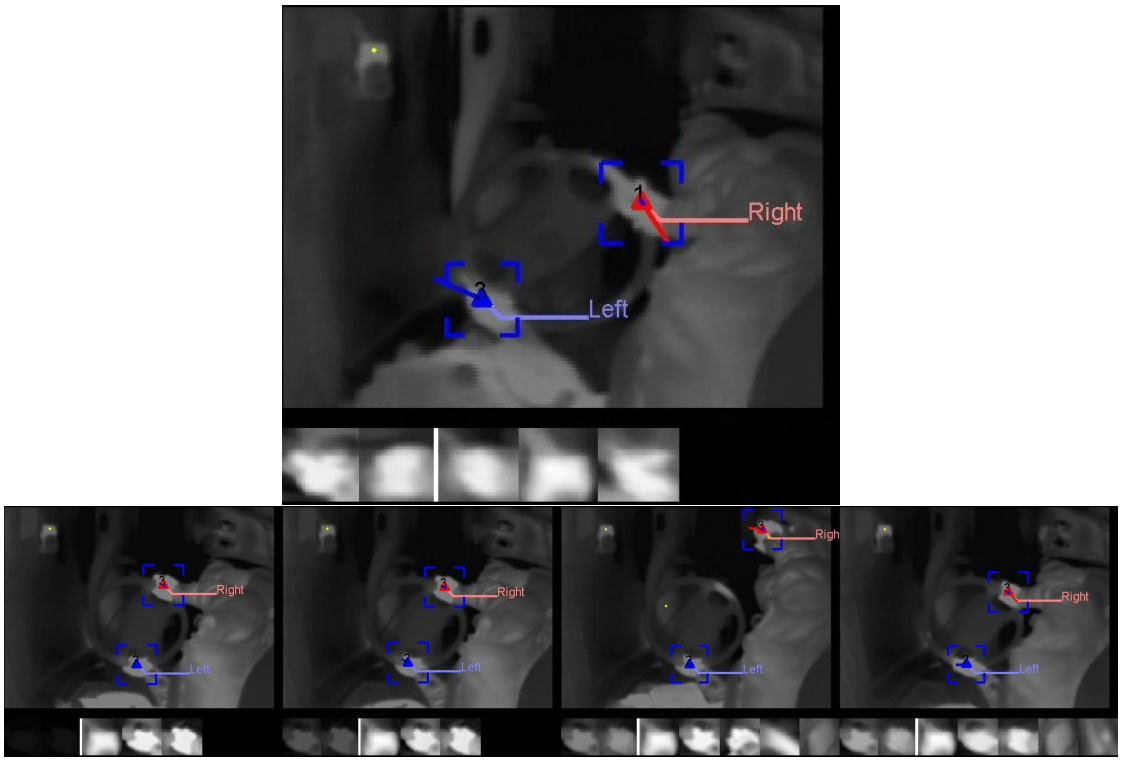


Figure IV.5: Progression of the first few frames of the hand tracking result. Beneath the main image are several image patches. They show the left and right hand appearance model and the track appearance models. The progression shows the left and right hand appearance model being updated over time relying on hand position prior.

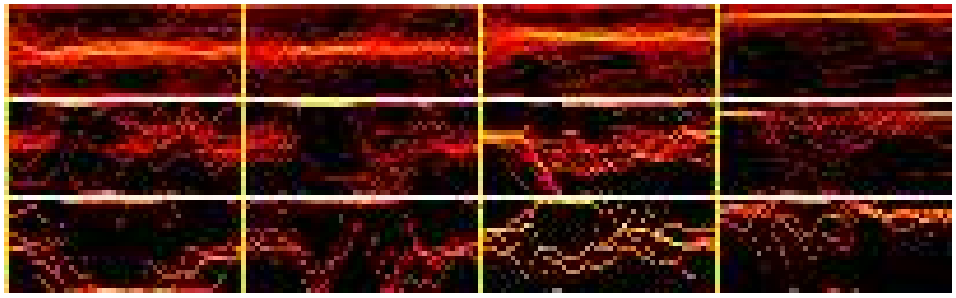


Figure IV.6: 2D trajectory histogram of the detected hand tracks time-aligned to the intersection-turn boundary crossing for the go-forward, turn-left, turn-right maneuvers (one for each row), showing the left-hand x, left-hand y, right-hand x, and right-hand y image coordinates (one for each column).

and shown in fig. IV.6. Sequences with a regular pattern of movement over time will appear in the 2-D histogram as a hot streak over time. The x- and y-axis of each histogram are time in samples and the position in the image. The first row consists of go-forward (gf) maneuvers. The second and third rows consist of the turn-left (tl), and turn-right (tr) sequences respectively.

A comparison of the three rows clearly shows distinct motion patterns of both the left and right hand during the 3 different activities. The comparison of the two leftmost and rightmost columns in a given row shows the similar transition patterns of the left and right hands with different offsets. This analysis uncovers specific hand motion patterns in each activity. The two hands' relative position and motion with respect to the steering wheel rotation constitutes five different types of grasp *operation triplets*, explained next.

IV.4 Hand Grasp Analysis

As the development of a thermal vision-based hand tracking algorithm progressed, it became clear that hand position with steering angle information can be combined to determine the various grasping activities of the vehicle. While the HMM-based driving activity classifier provides gross semantic-level recognition of driving behavior such as gf, tl, tr, the grasp analysis, supported by the hierarchical activity grammar introduced in [43], provides finer semantic-level behavior representation and analysis.

The hand tracker described in this chapter is used to provide the (x, y) positions and (v_x, v_y) velocities of the left and right hands. Steering angle information is combined with hand tracking information to recognize 5 hand grasping behaviors:

1. A hand grasps but does not move the steering wheel.
2. A hand grasps and moves the steering wheel in the counterclockwise (left) direction.
3. A hand grasps and moves the steering wheel clockwise (right).
4. A hand grasps the steering wheel loosely and allows the wheel to turn underneath it.
5. A hand does not grasp the steering wheel.

The grasping behavior is detected by measuring how correlated are the independently measured movements of the hands and the steering wheel angle. If the steering moves together with the hands and the hands are in proximity of being able to grasp the wheel, then the hand is presumed to be grasping the steering wheel. If neither conditions is met, the hand is presumed otherwise.

The operation-triplet is used to describe these grasping behaviors. Borrowed from linguistics, the operation-triplet consists of three elements: an agent, motion and target [43]. This representation can completely describe the activities of the hand. The agents in this case are the left and right hands. The target is either the steering wheel or null. The null target is used to describe the activity where the hand interacts with anything else besides the steering wheel.

To determine if the hand is grasping the steering wheel, an approximate ellipse model of the steering wheel is used to measure the distance between the hand and the steering wheel. This is accomplished by manually fitting an ellipse over the steering wheel in the LWIR image, such that all points $\mathbf{x}$ on the steering wheel in the image are a distance of 1.0 away from the center of the steering wheel. This is accomplished by using the weighted 2-norm

$$d(\mathbf{x}) = (\mathbf{x} - \mathbf{x}_o)^\top \mathbf{S} (\mathbf{x} - \mathbf{x}_o) \tag{IV.1}$$

where $\mathbf{x}_o$ is the center of the steering wheel in image coordinates and $\mathbf{S}$ determines the shape of the ellipse. A hand detected within a deviation $1 - \tau_d < \delta_{sw} < 1 + \tau_d$ for some τ_d is considered in position to grasp the wheel; a hand outside that range is considered performing other motions towards the null target.

To determine which of the five grasping maneuvers the hand is performing, the correlation of the steering wheel and hand angular velocities around the center of the steering wheel is examined. The angular positions of the hands around the steering column are found by calculating the angle

$$\theta = \arctan((y - y_o)/(x - x_o)) - \theta_o \tag{IV.2}$$

where $\mathbf{x} = (x, y)$ is the position of the hands in image coordinates, $\mathbf{x}_o = (x_o, y_o)$ is the center of the steering wheel, and θ_o is the bias applied to align the 0 degree position to the top of the steering wheel. The angular velocity ω_h is measured by taking the difference between the last and the current angle position of the hands normalized by the duration of time passed.

$$\omega = \frac{\theta(t) - \theta(t + dt)}{dt} \tag{IV.3}$$

Finally, we define the correlation between steering wheel and hand angular velocities around the wheel to be the difference $\rho = \|\omega_h - \omega_{sw}\|$.

For hands in position to grasp the wheel, the hand is determined to be performing one of the following three activities: If the difference between angular velocities of the wheel and hand exceed a threshold τ_ω, the hand is then determined to be grasping the wheel loosely, and allowing the wheel to turn underneath them. If the difference of angular velocities are within that threshold, the hand and wheel are determined to be moving together. If both the hand and wheel angular velocities are zero, the hands are grasping and resting on the wheel. Tab. IV.1 summarizes the test conditions under which the various operation-triplets occur.

The five grasp operation-triplets are illustrated in fig. IV.7 and IV.8.

IV.5 Discussion and Concluding Remarks

In this chapter, we presented a solution to the requirement of tracking hand positions in vehicles. We proposed the use of long-wavelength infrared imagery to capture the nearly constant temperature of

Table IV.1: Driver Hand Grasp Operation-Triplets

Grasp operation-triplet <agent,motion,target>	Conditions
<l/r hand, grasp+no move, sw>	$d(\mathbf{x}_h) - 1 \leq \tau_d$ $\|\omega_{sw} - \omega_h\| \leq \tau_\omega$ $\omega_{sw} = 0$
<l/r hand, grasp+turn left, sw>	$d(\mathbf{x}_h) - 1 \leq \tau_d$ $\|\omega_{sw} - \omega_h\| \leq \tau_\omega$ $\omega_{sw} < 0$
<l/r hand, grasp+turn right, sw>	$d(\mathbf{x}_h) - 1 \leq \tau_d$ $\|\omega_{sw} - \omega_h\| \leq \tau_\omega$ $\omega_{sw} > 0$
<l/r hand, grasp+sliding over, sw>	$d(\mathbf{x}_h) - 1 \leq \tau_d$ $\|\omega_{sw} - \omega_h\| > \tau_\omega$
<l/r hand, no grasp, null>	$d(\mathbf{x}_h) - 1 > \tau_d$

hands. Using a cascade of boosted classifiers and probabilistic multi-target tracking framework, we were able to take advantage of the stability of the driver's appearance in thermal imagery and demonstrate a system that is able to track both the left and right hands of the driver over a course of 90 minutes of driving of a single driver.

The results of the hand tracking was combined with steering information to determine 5 grasping activities using the operation-triplet construct. These 5 grasping activities compactly describe whether either hand is grasping the wheel or not. And if grasping, whether the hands are actively turning the wheel to the left, to the right, not turning, or allowing the wheel to spin underneath.

The text of Chapter IV, in part, is a reprint of the material as it appears in: Shinko Y. Cheng, Sangho Park, Mohan M. Trivedi, "Multi-spectral and Multi-perspective Video Arrays for Driver Body Tracking and Activity Analysis," Computer Vision and Image Understanding: Special Issue on Advances in Vision Algorithms and Systems Beyond the Visible Spectrum, vol. 106, number 2–3, pages 245-257, May-Jun. 2007. Sangho Park and I were the primary researchers of the cited material, and Professor Trivedi directed and supervised the research which forms the basis of this chapter.

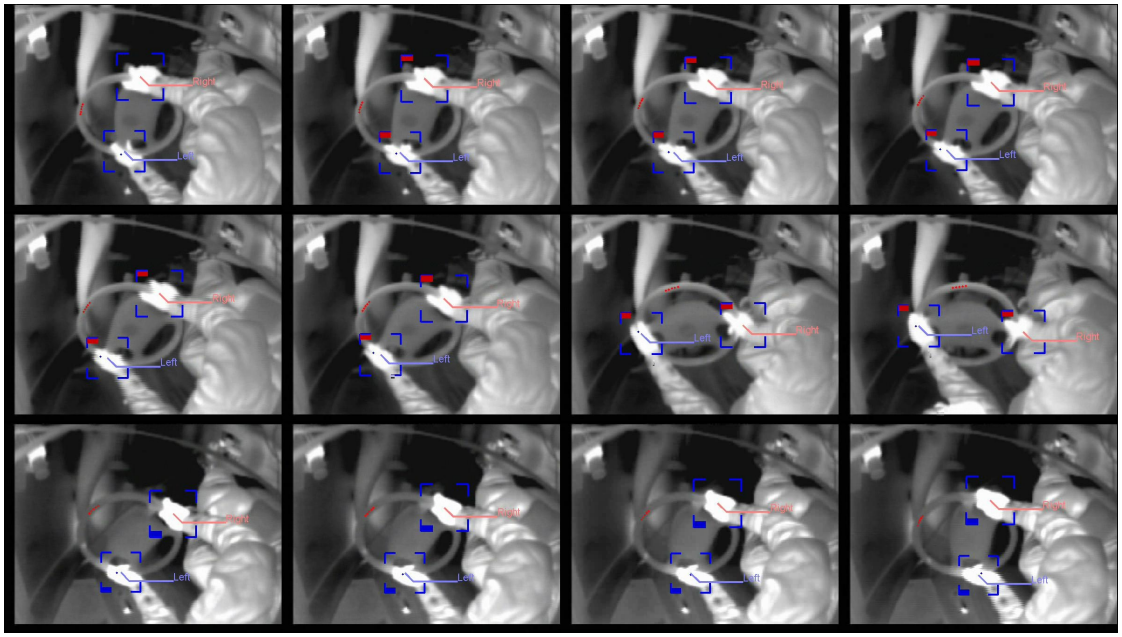
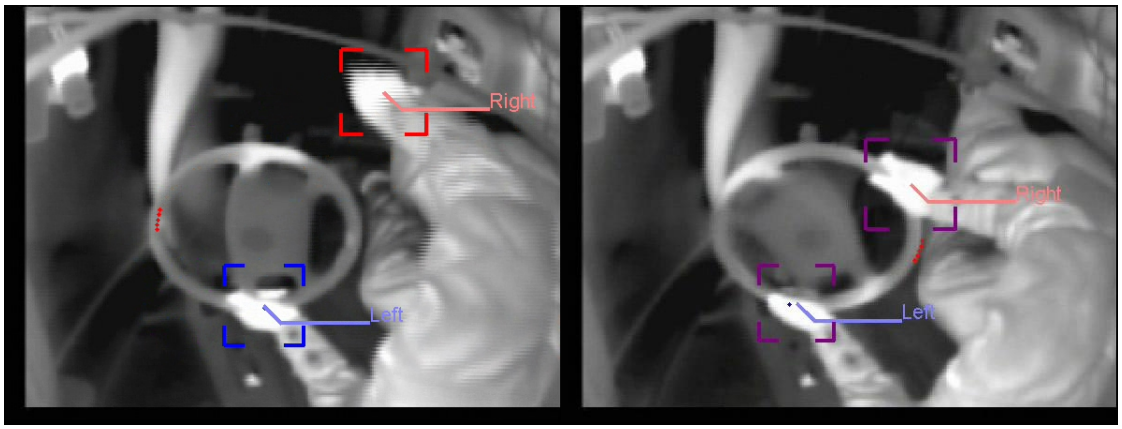


Figure IV.7: This figure illustrates the two grasp types: Hand on wheel turning right and left. The red and blue bricks in the bounding box represent a hand is moving the steering wheel right and left, respectively.



(a) (b)

Figure IV.8: Illustrated are three more grasp types: (a) Left hand is grasping the wheel holding still (in blue) while right hand is away from the wheel (in red). (b) Both hands grasping but slipping over the steering wheel while the steering wheel straightens itself out (in purple). The grasp activity is found for one hand independently of the other.

V

In-Vehicle Vision-based User Determination

Knowledge of driver body pose can be used in many applications. In this chapter we present a novel robust computer vision algorithm for discriminating which of the front-row seat occupants is accessing the infotainment controls. The information content is intended to alleviate driver distraction and maximize passenger infotainment experience.

V.1 Introduction

A broad new array of devices is finding its place in today's vehicles. The infotainment device has graduated from a term referring to the radio to a collective word to describe the navigational, vehicle status view, climate control, personal cell-phone control, MP3 player control, and web-browsing, and even television functionalities of the front console area of the vehicle [44, 45]. With all of these opportunities for drivers to be distracted, the solution has been to limit the functionality or the output from these infotainment systems, and make them less distracting. Often, the information provided by these devices becomes oversimplified and not rich enough for passengers who are not required to maintain attention to the aspects of driving. It is far more desirable to alleviate driver distraction yet at the same time allow occupants of the vehicle access to better information.

We therefore propose a novel Vision-based User Determination (VUD) system to determine which front-row seat occupant is accessing the infotainment controls. This device is intended to simultaneously improve the safety of the vehicle by alleviating driver distraction from the vehicle's infotainment system, and allow the passenger full access to the information device. The controls of the infotainment system consist of buttons and a knob with rotational and directional degrees-of-freedom. They are as-

sumed located in the aisle area between the driver and the passenger.

Vision systems afford additional functionalities with a single sensor. From the perspective of sensor efficiency, an imaging device with a good wide perspective can sense a variety of useful information about the front-row seat environment, e.g. tracking hands.

The challenges of developing such a system center on developing a robust classification algorithm that is capable of maintaining high performance in all operating conditions of the vehicle. That is to say, good performance should be maintained through changes in the appearance of people, changes in lighting from different times of the day, and differences in the camera position during installation (translation). Because vehicles are likely to receive maintenance only between several months of operation, if at all, much of the functionality must also require little or no maintenance. These attributes were achieved with appropriate choices in the design of the pattern classifier and system components.

The proposed module takes as input visible and near-infrared images of the front-seat and center-console area illuminated with a bank of near-infrared LEDs. The module then uses these images to determine which front-row occupant is accessing the device, if anyone at all. The histogram of oriented gradients (HOG) image descriptor was chosen to create the feature vectors [46]. The module then utilizes the kernel support-vector-machine (SVM) to classify the observed image features into the three classes.

The evaluation of this approach uses 2 metrics: the correct classification rates of each class forming the confusion matrix, and the average correct detection rate of the three classes. In the training process, care was taken to ensure that a representative data-set was used, and the usual cross-validation techniques were employed to gauge the generalizability of the pattern classifier. Data was collected at 4 different times of day with 8 different individuals for a total of 18 test-runs, over 1-hour of data at 30 observations per second. A large representative data-set allows for an understanding of the performance on a wider range of operating conditions. We also analyzed the system's invariance to translation in the x- and y- directions of the image patch, where the features are extracted. These qualities influence the flexibility in camera placement during the installation process.

The trained system can correctly recognize whether the driver, passenger, or no one has their hand over the infotainment controls with better than 95% average correct classification rate. This rate is the average percentage of each category that was classified correctly.

V.2 Related Work

The idea of tailoring vehicle information system functions, input and output devices, and user interface based upon whether the user is the driver or passenger is not a new one. Chou *et al.* [47] have proposed the use of weight sensors to determine the presence of a passenger before enabling full-functionality

of an infotainment system. Harter *et al.* [48] too proposed to switch between "enhanced functionality" and "base functionality" of the information system by determining the presence of a passenger but used proximity sensors instead. They take a step further to determine when to engage "base functionality" by determining whether the driver has gazed into the infotainment monitor more than 2 seconds (considered too long) using a vision-based eye gaze tracking system. We propose to use the same vision modality but analyze the hands of the occupants rather than the driver's head to determine when to switch between functionality modes. Our proposed solution can replace or complement these other systems by providing the following advantages:

1. The proposed system is arguably simpler to implement and maintain than an eye gaze tracking system. The proposed solution requires no camera nor person calibration.

2. The proposed system actively monitors the hands to detect the intent to access the information system as soon as the hand nears the controls.

3. The proposed system actively detects the null case, i.e. when no one is accessing the information system. This case can be used to automatically show and hide access controls in the display. This cases can be construed as having a more attentive driver and require less driver assistance. Weight sensor and proximity sensor systems cannot detect the no-one case.

4. The proposed system can detect difficult situations with occlusion, including the partial occlusion from the other occupant's hand.

Tab. V.1 summarizes the related work in user determination systems.

The problem of hand image based user determination can approached in two ways: 1) Active tracking of occupant hands as the hand passes into and out of the area over the infotainment controls (or region-of-interest) to detect intent to access. 2) Learn the appearance of the driver's hand, the passenger's hand or no one's hand over the region-of-interest. A number of works have addressed the first approach.

The first type consists of a detector which locates the hand in the images, and then tracking the hand. Tracking is associating one hand detection in time with the next. The challenge of this approach is in obtaining a good description of the appearance of the hand in its various poses, and a way to efficiently check all areas of the image for the existence of a hand. The characteristics of a good descriptor is one that would correctly associate two hand detections of the same hand in different positions and poses.

One hand detection algorithms devised is by Kölsch *et al.* [49] employs a cascade of boosted classifiers using haar-wavelet-like image features and their extensions to determine whether an image patch, among all possible patches in an image, consisted of a hand or not. The rates reported for real-time operation were very good (92% detection rate with a false positive rate of 1e-8), but the approach detected hands in a standard canonical position: fingers up and thumb to the right and 7 other similar forms. Kölsch

Table V.1: Related work on User Determination for Information System Mode Control.

	Objective	Method and Result	Cues used	Comment
Chou *et al.*('99) [47]	User Discrimination Control of Vehicle Information Systems	Occupant Presence using weight sensor.	Weight sensor	-
Harter *et al.*('02) [48]	User Discrimination Control of Vehicle Infotainment System	Prolonged Driver Eye Gaze Detection Passenger Presence using multi-modal sensors	Weight sensor Driver eye-gaze sensor Proximity sensor seat-belt tension sensor	-
Proposed Work	Determine whether Driver/ Passenger/ No-one's hand is present.	HOG feature, 3-class SVM classifier, 97.8% Avg. CCR	Hand imaging sensor	Analyzed sample-by-sample correct detection rate.

*et al.*proposed using a flock-of-features approach to track the positions of the hands after initial detection, and to address the problem of maintaining track of hands through its many poses.

We employed a similar detection technique with long-wavelength thermal infrared images of hands (chap. IV) [37]. The thermal infrared modality is especially appropriate in the vehicular domain because image appearance is not at all affected by changing illumination conditions and pixel intensity of skin is stays relatively constant. Because of the special quality of hands in thermal images, this detector was also effective in detecting hands in various poses. The detector is applied on each in-coming frame and the multiple hands are tracked using the Kalman filter and Probabilistic Data Association Filter (PDAF) to disambiguate one track from another. This approach however suffers from the use of a thermal camera, which are still expensive as compared to the visible-wavelength camera.

Yuan *et al.* [50] proposed a hand tracking system that utilizes color and motion information to detect and the Viterbi algorithm to track the hands. Because of the vehicle's high and low light operating conditions and the need to discreetly illuminate the vehicle interior, color cameras — and therefore skin-color based algorithms — were not an option.

We address the user determination problem using the second approach, which is the direct classification of images of the infotainment controls region. No tracking is required, although some rudimentary filtering of the classification responses over time will increase the correct classification rate. This approach takes advantage of the fact that the region-of-interest has a stable background which is of the vehicle interior. To the best of our knowledge, no other work approaches the user determination

problem in a similar way.

V.3 Pattern Classification Considerations

The core of the VUD system is a pattern classifier trained from annotated examples. A sensor observes a hand over the infotainment controls, and the task is to determine among the occupants of the vehicle to whom the hand belongs.

The typical structure of a pattern classifier consists of the following stages:

- Sensing - Makes observations of the nature to be classified or categorized.
- Feature Extraction - Efficiently describes the raw observations by retaining only the information that is useful for classification and composing the description in the form of a feature vector.
- Classification - Assigns the feature vector to a class or category.

Functionally, if $\mathbf{x} \in \mathbb{R}^d$ is the d-dimensional feature vector, then the classifier g transforms $\mathbf{x}$ to the intended target value, or class $\hat{t}$, given by

$$\hat{t} = g(\mathbf{x}, \theta) \tag{V.1}$$

where $\hat{t} \in \{C_1, C_2, ...\}$, and θ consists of the set of parameters to be optimized, or "learned", through classifier training. "Supervised learning" is training performed by presenting a set of feature vectors with known class assignments, assigned by an expert or human observer. From a slightly different perspective, a classifier can be described as creating decision boundaries in the d-dimensional feature space. Each feature vector has an associated class label. The task of training the classifier is to create decision boundaries in this feature space such that the resulting regions contain only the observations (including never before recorded observations) with the same class label. If there exists a hyperplane that can separate two classes of features, the features are said to be linearly separable.

The factors that impact the success of a pattern classification system are: 1) The choice of the classifier (model), 2) the training examples, and 3) the method of feature extraction.

A good choice of a classifier is one that will create decision boundaries that represent the true boundaries in nature. For example, if the feature space is nonlinear, the model must be able to produce nonlinear decision boundaries. The support-vector-machine (SVM) with an appropriate kernel function is a popular technique successfully used in many applications, including optical-character-recognition and finger print identification [38].

At its center, SVM is a two-class classifier that creates a hyperplane that separates the two classes of features with the greatest margin or distance between the feature and the hyperplane. By itself, SVM solves the classification problem when the features are linearly separable. A classification is made

by taking the sign of the value resulting from the linear combination of the elements of the feature vector with the learned weights ($g(x) = \text{sign}(\mathbf{a}^\top \mathbf{x})$). Its success in applications stems from SVM's ability to learn sparse solutions for the weights when the feature vectors are projected onto a higher-dimensional space using kernel functions. Kernel functions emphasize certain aspects of the spatial configuration of the features in feature space, such as the distance between example feature vectors for the data-driven Kernel function, and making the otherwise complex non-linearly-separable features into linearly-separable ones. The sparseness of solutions results in efficient on-line classification. For details on the learning algorithm for SVM, we refer the reader to many good texts [38,51,52]. More details on the SVM are also described in the next section, sec. V.4, when the actual the VUD system is described.

The second factor that influences the success of a pattern classifier is ensuring that the example observations presented to the classifier are representative of the kinds of observations gathered where the classifier is deployed. The analogy is that if a person is asked to pick out Fuji apples from a case of Fuji and Red Delicious apples having only learned about apples and oranges, the person may mistakenly pick out all the apples. Presenting the classifier with every possible kind of observation will ensure that the classifier is trained to react correctly in these more ambiguous situations. The examples in our data-set were collected from various times of day with various people in both the passenger and driver positions. The data-set also contains a sequence of a variety of clutter. We describe how the data was collected in sec. V.5.

There is also the possibility that two classes are too similar or the feature vectors from different classes have considerable overlap in the feature space. Such a feature space is not perfectly separable by any division of regions. This may then indicate that a different description of the observations should be explored in the feature extraction phase.

This brings us to the last factor that influences the success of a pattern classification system which is feature vector construction. The histogram of oriented-gradients (HOG) and other gradient orientation histogram-style local descriptors have been shown to be highly effective in characterizing the appearance and shape without the need for precise positioning of the gradient or edge positions in the image interest point [53, 54]. This type of local image descriptor has been shown to be very effective in finding corresponding points between two images where one image is a transformed version of the other. The SIFT and GLOH descriptors, which both have elements of the basic HOG descriptor, were shown to be the most successful in matching points in images with very different perspectives. These descriptors extend the HOG descriptor by adding rotation and shift invariance. The HOG component itself gives the descriptor robustness to change in perspective, illumination level, and even focus. We chose to describe the appearance of the infotainment controls with the HOG descriptor for these reasons. By a thorough evaluation below, we determined that the histogram of oriented gradients feature descriptor is adequate.

Table V.2: VUD hardware specifications

Device	Attribute	Value	Notes
Imager	Make	VidereDesign	
	Model	STH-MDCS2-VAR	
	Image Resolution	640x480 px, 8bits/px	1280x960 px max
	Frame Rate	30 Hz	Max 7.5 Hz at 1280x960
	Image Type	400-700nm monochrome	visible spectrum
	Sensor Type	$\frac{1}{2}$ in. CMOS	
	Lens focal-length	3.5 mm	C/CS mount
	SNR	> 45dB, no gain	
	Sensitivity	2.1 V/lux-sec	
	Power	2 W	
	Size	1.5H x 2.6L x 1D in.	without lens
	Weight	425g (15 oz)	without lens
Illuminator	Make	SUPERCIRCUITS	
	Model	IR14	
	Number of LEDs	140	
	Configuration	Planar grid array	
	Power	12-15V 1A	

V.4 Vision-based User Determination System

The Vision-based User Determination (VUD) system determines the individual whose hand is accessing the infotainment device by classifying patches of captured images of the front-row seat area in a passenger vehicle. The user is defined as one of three categories: 1) driver, 2) passenger, and 3) no-one. The infotainment controls are assumed to be positioned just aft of the gear-shift, forward of the hand-rest, and beside the hand-brake.

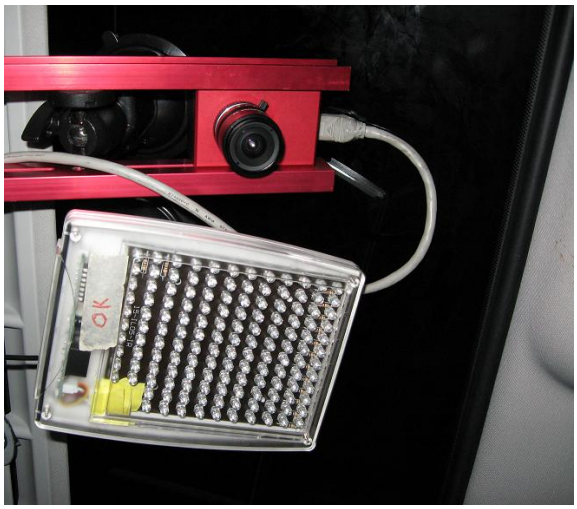
We adopt the visible and near-infrared spectrum imaging modality to provide the observations for determining whose hand is on the infotainment controls. The primary reasons are the passive nature of the camera; at night, the front-row seat area can still be captured by illuminating the area with near-infrared illuminators without distracting the occupants. Example images are shown in fig. V.1(a) and V.1(b). Hardware specifications are listed in tab. V.2.

The overall system has three stages: data capture, feature extraction, and classification. This is the basic procedure for all pattern classification systems.

The system starts with the capture of monochrome images. A rectangular image patch that spans between the edges of the driver's and passenger's seat and the length between the gear-shifter and the hand rest is extracted. An example image captured from the front seat area and the image patch are shown in fig. V.3. The histogram of orientation gradients (HOG) description of the image patch is calculated, and then presented to the multi-class kernel support-vector-machine (SVM) classifier to determine which of 3 events occurred: the driver's hand, passenger's hand, or no one's hand accessed the infotainment controls.

(a) Example image in daylight.

(b) Example image at night.



(c) Camera and illuminator.

(d) Camera and illuminator set-up.

Figure V.1: Example images captured during the day and night, and the positions of the camera and illuminator in the LISA-P test-bed for the VUD system.

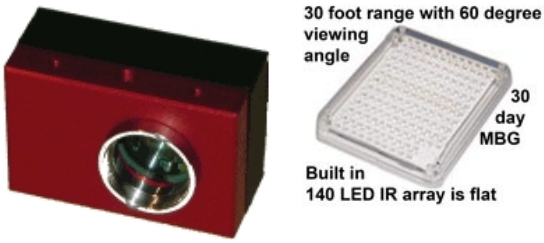


Figure V.2: VidereDesign STH-MDCS2-VAR camera and SUPERCIRCUITS IR14 140 LED IR Illuminator were used for the VUD system.

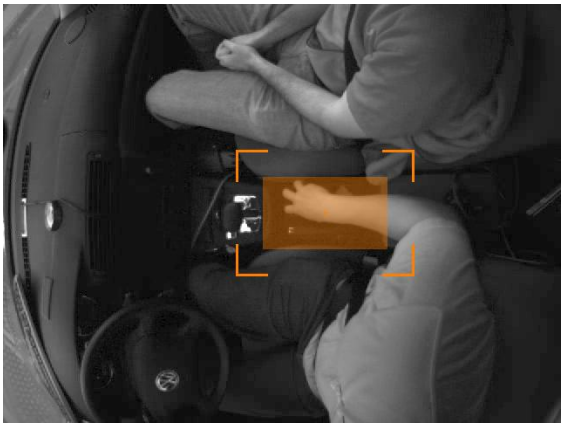


Figure V.3: Image region-of-interest used to determine user in the VUD system.

Table V.3: Parameters used for the histogram of oriented gradients feature descriptor in the VUD system.

Parameters	Value
Bins along x-axis	2
Bins along y-axis	2
Bins along orientation	8

V.4.A HOG Feature Extraction

The HOG descriptor for an image patch is created by first taking the gradient of the patch. The resulting gradient image is then divided into smaller rectangular patches of pixels specified by the number of x-bins and y-bins in the x and y directions. Within each rectangle, a orientation histogram is generated from the pixels contained within each smaller rectangular patch. In generating this orientation histogram, the number of o-bins specify the number of divisions along the span of gradient orientations (0 to 360 degrees) The o-bins parameter also specifies the length of each histogram for each rectangle. All the orientation histograms are then vectorized and concatenated to form the feature vector $\mathbf{x}$. Altogether, 3 parameters determine the dimensions of the final feature vector: the number of divisions along x, y, and the number of bins in the orientation histogram. For example, a 2x2 grid of bins with 8 slices in the range of possible gradient orientations results in a 32-dimensional feature vector (2x2x8) for each image patch.

V.4.B SVM Classifier

The objective of any classifier is to correctly assign the observed feature vector $\mathbf{x}$ to its corresponding label or class k of K classes. Mathematically, this refers to creating a set of discriminant functions $g_i(\mathbf{x})$ for $i \in 1, ..., K$ such that $g_k(\mathbf{x})$ produces the highest value when $\mathbf{x}$ corresponds to class k. The discriminant function is parameterized by a set of variables represented as θ. Training a classifier refers to optimizing the parameters θ such that the classifier will correctly classify as many input features

vectors $\mathbf{x}$ as possible in a given training data-set. The data-set contains example feature vectors and their associated class values or target values, manually assigned.

The support-vector-machine was chosen to be the underlying classifier for this application. The SVM classifier takes the form

$$g(x) = \text{sign}\left(\sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b\right) \tag{V.2}$$

where $t_n \in \{-1, 1\}$ is the target value for feature $\mathbf{x}_n$, while a_n and b are the weights to be optimized. The key result of SVM is the sparse solutions for a_n, i.e. many terms are zero. The effect of this is efficiency in classification; only a very small subset of N examples actually need to be retained to calculate $g(\mathbf{x})$. The feature vectors $\mathbf{x}_n$ for which the corresponding a_n is non-zero are also called the "support vectors."

The vector of coefficients a_n can be seen as a hyper-plane (n-dimensional plane) separating the two sets of features (one corresponding to $t = -1$, and the other to $t = +1$) in the feature space. The optimal condition is when the two sets of features are separated by this hyper-plane with the largest possible spacing, or margin, between the hyper-plane and the closest feature vectors in this feature space. Specifically, the objective function that is maximized over $\mathbf{a}$ is

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{M} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \tag{V.3}$$

subject to the constraints

$$0 \leq a_n \leq C \tag{V.4}$$

$$\sum_{n=1}^{N} a_n t_n = 0 \tag{V.5}$$

where $n = 1, ..., N$. This is an example of a quadratic programming problem in which we are trying to minimize a quadratic function subject to a set of linear inequality constraints. For a more detailed treatment of the SVM, please refer to [38, 51, 52].

SVM by itself is a two-class classifier. Multi-class SVM is used to classify the feature vectors into the 3 classes for the VUD system. This extension is achieved by training 3 SVM classifiers with the one-versus-rest approach. Each two-class SVM classifier will be trained with the feature vectors annotated as one class with a target value $t = +1$, and the other feature vectors grouped together with target value $t = -1$. There are three two-class SVM classifiers in all. Each SVM classifier can be seen as one of 3 discriminant functions as defined in equ. V.1, and the final classification is done by determining which of the 3 functions yields the highest value, i.e. determine which of 3 regions in feature space does the feature vector lie in the deepest. Formally, the classified result C is given by

$$C = \arg\max_{C_k} g_{C_k}(\mathbf{x}) \tag{V.6}$$

Because the raw features do not neatly lie in their respective sides of this hyper-plane in the raw feature space, i.e. the raw features $\mathbf{x}$ are not "linearly separable," we project these features onto a higher-dimensional kernel space $k(\mathbf{x}_i, \mathbf{x}_j)$. Depending on the chosen kernel function, different aspects of the spatial configuration of the raw features are emphasized by the kernel function. The radial-basis-function (RBF) for example is one type of kernel function where each kernel function depends on the distance (usually Euclidean), from a specified mean μ_i. The means are set to the feature points, producing as many kernel functions as there are example feature points. The use of the kernel function effectively projects the raw feature space from a d-dimensional space onto an N-dimensional space, where $d << N$. Arriving at a sparse solution where only a small subset of N is retained as part of the classifier is the problem that SVM solves.

The proposed system was prototyped with the SVM implementation in OpenCV Machine Learning Library.

V.5 Experimental Evaluation

The experimental evaluation consists of the definition of relevant performance metrics, validation of the performance evaluation as being representative of the true performance, validation of the optimality of the algorithm parameters, and finally discussion of the results pertaining to robustness of the system in conditions in which they may fail.

V.5.A Performance Metric

The evaluation of the VUD system utilizes 2 metrics: 1) the confusion-matrix summarizing the classification rate of a feature vector of a given class as a given class, and 2) the average correct classification rate among the three classes. The confusion matrix consists of 3 rows and columns for each of the 3 classes. The row represents the actual class of novel examples (excluding examples used for the training of the classifier), and the columns represent the predicted category of those examples. Perfect recognition will yield a confusion matrix with zero values in the off-diagonal elements. A normalized confusion-matrix represents the percentage of, or the probability that a particular class will be predicted as one of the classes. The normalized confusion-matrix is calculate by dividing each element of the row by the sum of that row in the confusion matrix. Worst performance is considered the performance that can be achieved by random guessing, which generates a normalized confusion matrix with 33% classification rate in each matrix element.

V.5.B Validation of Performance

During the training of any pattern classifier, there is a risk of over-fitting, where the classifier is trained to the point when the training set is classified perfectly, but when presented with novel examples the correct classification rate is worse than can otherwise be achieved. This is due to noise associated with overlapping class regions in feature-space. A classifier that has been over-fitted would create decision boundaries to classify these noisy samples correctly and thereby degrade classification performance for non-noisy samples.

To address this, 5-fold cross-validation is used to estimate the expected recognition rate, a better indicator of the generalizability of the pattern classifier to as-yet-unseen data. That is to say in order to generate a performance measure that represents more closely the true classification performance on as-yet-unseen data, we use cross-validation to estimate the correct classification rates. The data-set was divided into 5 sub-sets. A multi-class SVM classifier was trained on all but 1 of the 5 sub-sets of examples (frames), and the classification rates are calculated from the remaining sub-set to produce 5 normalized confusion matrices. The average of all 5 recognition rates are found and reported. The standard deviation of each element in the confusion matrix is also found and were always less than $.5\%$ difference.

To ensure that the trained results would perform well in real situations, the data-set was collected at various times-of-day (noon, afternoon, twilight, night) with various individuals (8 individuals) in both the driver and passenger position. One sequence was captured with a variety of clutter (flashlight, card-board, paper, mouse-pad, tools, cups) introduced into the region-of-interest to capture the statistics of feature vectors of those instances. A total of 18 video sequences containing a 114,886 examples and 63 minutes of video in various illumination conditions were used for training and testing. The conditions under which the data-set was collected is summarized in tab V.4 and V.5. The people present in the data capture are denoted as A through H for the 8 people. An asterisk denotes no occupant for that video sequence. Individuals wore short and long-sleeve shirts. For most sequences, the data was captured while the vehicle was in motion, driven along a route in which the direction of sunlight could shine into the vehicle from every direction at least once. Different times of day yielded different angles of the sun and character of the sun.

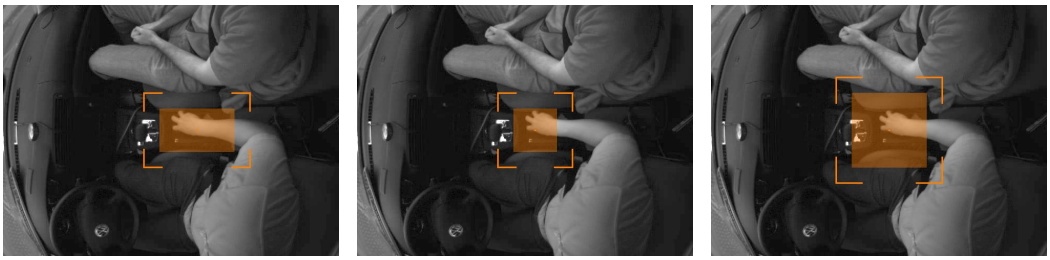
Each frame of the video is manually annotated with the category to which it belongs. Namely, each frame may show either no one, the driver, or the passenger is placing their hand over the infotainment controls area. There are a total of 68,467, 20,179 and 25,340 unique frames collected for each of the three classes, respectively.

Table V.4: Summary of attributes of the 18 sequences of video data used for training and testing for the VUD system. Test subjects are labeled A through H, for 8 subjects in all.

Group	Seq.	Frames	Occupant and Position (Driver/ Passenger)	Vehicle State	Weather Condition	Time-of-Day
1st	1	4814	A/*	Stationary	Indoor	N/A
	2	6000	*/A	Stationary	Indoor	N/A
	3	6947	A/*	Stationary	Overcast	6pm
	4	4089	*/A	Stationary	Overcast	6pm
	5	11,740	A/B	Moving	Overcast	7pm
	6	7093	C/A	Moving	Sunny	12pm
	7	7699	D/E	Moving	Sunny	12pm
Group Total		48,382	Examples in each class: {31,963, 8650, 7769}			
2nd	8	13,012	A/A	Moving	Night	9pm
	9	4978	B/G	Moving	Sunny	12pm
	10	4202	G/B	Moving	Sunny	12pm
	11	6908	C/A	Moving	Sunny	12pm
	12	5445	A/C	Moving	Sunny	12pm
	13	4845	A/*	Moving	Sunny	12pm
	14	5039	H/G	Moving	Sunny	4pm
	15	5002	G/H	Moving	Night	4pm
	16	6961	H/A	Moving	Night	9pm
	17	3987	A/H	Moving	Night	9pm
	18	6125	A/H	Moving	Night	9pm
Group Total		65,604	Examples in each class: {36,504, 11,529, 17,571}			
Total		114,886	Examples in each class: {68,467, 20,179, 25,340}			

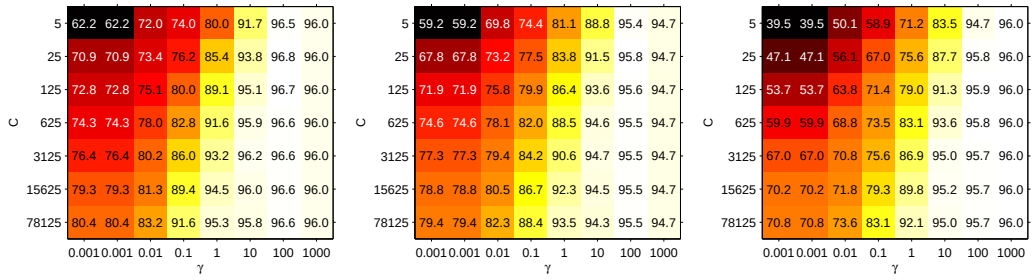
Table V.5: Summary of users.

Class	Description	Examples
1	No One	68,467
2	Driver	20,179
3	Passenger	25,340



(a) Exp 1 (b) Exp 2 (c) Exp 3

Figure V.4: Various image patch sizes were used in evaluating the VUD system.



(a) Exp 1 (b) Exp 2 (c) Exp 3

Figure V.5: In training the multi-class SVM for user determination, optimal values of C (slack parameter) and γ (RBF kernel width) need to be found. This was done by grid-search and the average correct classification rate is the quantity to maximize. Randomly selected 5000 samples of each class were used to train the classifiers for this grid-search. A total of 15,000 samples was used.

V.5.C System Parameters Optimization

Three feature types were analyzed to validate our choice of image patch dimensions. Intuitively, the forearm is a good indicator of whose hand is accessing the infotainment controls. A rectangular image ROI of size 140x80 as depicted in fig. V.4(a) appears to capture both the hand and the forearm compactly. The other two image patches consisted of a square image patch of sizes 80x80 and 140x140, centered around the hand as shown in fig. V.4(b), and V.4(c).

The multi-class SVM classifier with the RBF kernel has 2 parameters that require tuning: the slack parameter C, and the RBF kernel width γ. This is done by searching a grid of values for the optimal tuple (C, γ) that yields the highest average correct detection rate (mean of the diagonal elements of the normalized confusion matrix). A subset of the complete data-set was used to efficiently generate the values in the grid-search: 5000 examples of each class were randomly selected for a total of 15,000 examples in the new training set. The results the grid-search for all three feature-types is shown in fig V.5. The result from this search indicates that the optimal values are $C = 25$ and $\gamma = 100$.

The results of the parameter optimized kernel SVM classifiers are summarized in tab. V.6. The differences in percentage points were subtle, but the rectangular image patch produced the best results of the three different sized patches with a correct detection rate of 95.42%, 97.6% and 97.3% for the instances when no-one, driver, and passenger was accessing the controls, respectively. The columns of the confusion-matrix represent the instances of each predicted class while each row represents the instances in an actual class.

To give a better sense of the performance as a function of time, duration of time in error were calculated. The amount of time when the system was in error in one hour is calculated by multiplying the percentage of time in error by 60 minutes. The average number of minutes in error in one hour for the three types of features were 1.926, 2.544, 2.526 minutes respectively. There is an improvement in confining the image patch to a rectangular area of the aisle by 0.5 seconds on average. Of course, the error was not uniformly distributed over all time. Most of the error clustered together during the transition regions, described next.

During the annotation of the data, care was taken to ensure that a consistent strategy was used for when a hand is in transition to and from the image patch region. Usually, the percentage of the hand remaining in the image patch is used to determine whether or not the hand is still in the region. There may still exist some inconsistencies in the transition frames; some annotated as hand still in the region when in fact not, and vice versa. Furthermore, in transition frames, the hands are often blurred.

To determine how much of the inconsistency adversely influences the performance, we examine the proportion of errors which exist in the transition region. A transition region is defined as $\pm L$ samples surrounding the point of transition in the annotation file. The width of the transition region is $2L + 1$. As expected 50% of the errors occur within 0.5 seconds of the transition. This means that the most confusing frames are when the hands enter and exit the image patch region. The proportion of errors taper off as the transition window increases as shown in fig. V.6. The implication is that 50% of errors can be avoided by utilizing a delay before deciding on the presence of a hand.

In light of this, median filtering of the classification responses over time and a delay to the responses was introduced. The average correct detection rate was then remeasured. The rates for various median filter window widths were used and the resulting average correct classification rates vs. median filter lengths is shown in fig. V.7. A window of 0.63 s in width (19 samples) gave the best average correct detection rate of 97.9%, or 1.257 minutes per hour in error, an improvement of nearly 1 minute worth of errors.

V.5.D Robustness

The VUD system's invariance to translation in the x- and y-axis directions was also analyzed. These qualities influence the flexibility in camera placement during the installation process. The image

Table V.6: Summary of VUD performance. The performance is described using the confusion matrix and average proportion of 1 minute in error. The classifiers are multi-class kernel SVMs (C=25, RBF, Gamma=100) and the confusion matrices are averages of 5 training results via 5-fold cross validation. 5000 randomly selected examples per class were selected for each training.

(a) Exp 1

	P(predicted|actual)			
	NoOne	Drver	Psngr	
NoOne	95.42	2.01	2.57	
Drver	1.82	97.63	0.54	
Psngr	2.12	0.57	97.31	
Average Correct Classification Rate				96.79%
Average Minutes per Hour in Error				1.926 mins

(b) Exp 2

	P(predicted|actual)			
	NoOne	Drver	Psngr	
NoOne	94.98	2.44	2.58	
Drver	2.66	96.41	0.93	
Psngr	3.02	1.08	95.90	
Average Correct Classification Rate				95.76%
Average Minutes per Hour in Error				2.544 mins

(c) Exp 3

	P(predicted|actual)			
	NoOne	Drver	Psngr	
NoOne	93.73	2.77	3.49	
Drver	2.40	97.18	0.42	
Psngr	2.94	0.59	96.47	
Average Correct Classification Rate				95.79%
Average Minutes per Hour in Error				2.526 mins

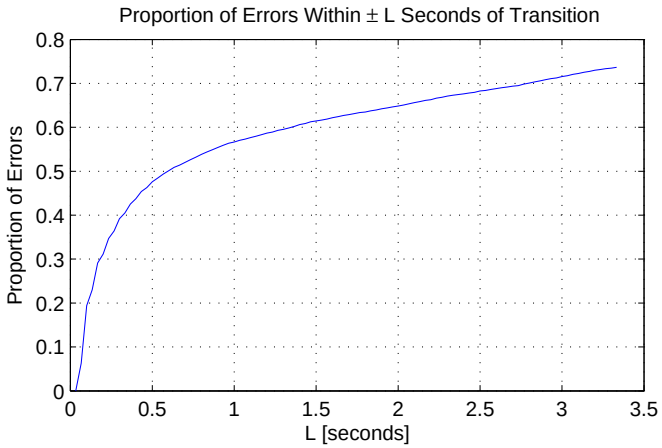


Figure V.6: Errors in the VUD system are examined to determine proportion of errors in the transition times, which are when the hands of the occupants enter or exit the image patch region.

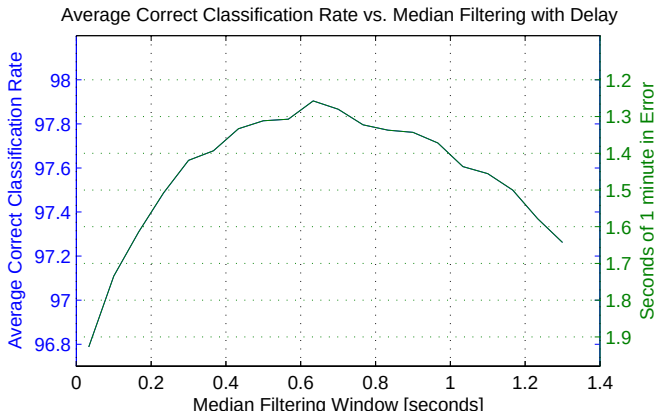


Figure V.7: Average Correct Classification Rate vs. median Filtering with Delay. Applying a smoothing median filter yields better correct classifications rate with an optimal window of .63 seconds in width. The corresponding delay is 0.235 seconds. The frame-rate is 30Hz.

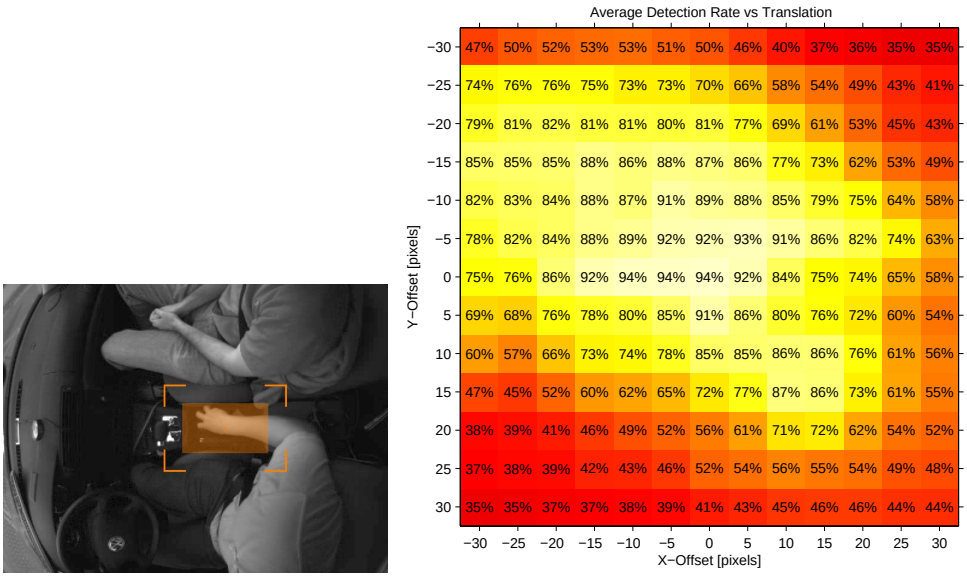
patch region-of-interest (ROI) is specified as shown in fig. V.3, but repeated for convenience in fig. V.8. This ROI was shifted to various positions in the image ± 30 px along the x and y directions. The range of translations is depicted in fig. V.8(a). The effect of those translations on average correct classification rate is shown as a heat image in fig. V.8(b). The average correct classification rates are collected from non-training frames from sequence 5 (see tab. V.4).

The results show that using the rectangular ROI, the performance of the VUD remains above 85% at ± 5 px deviation from the original location, and above 80% at ± 10 px. The latter amount of deviation amounts to approximately half the length of the occupant's finger.

V.6 Discussion and Concluding Remarks

We presented a vision-based user determination system. The system consisted of a visible and near-infrared imaging device observing the front-row seat area in the vehicle. Using histograms of oriented gradients features to describe the area over the controls, a support-vector-machine was shown to be able to provide 97.9% average correct classification with median filtering. With an offset of 10 pixels in any direction, the rate could still be maintained at better than 80%.

The system is intended to improve the safety and comfort of the vehicle by enabling the vehicle to determine which occupant is accessing the vehicle's infotainment controls, often characterized as one of the more distracting elements in a vehicle. It is a safety device in the sense that the vehicle would know whether there is a potential of the driver to be distracted in a critical situation, adding one more piece to the puzzle of automatically determining the driver's situational awareness. It is a comfort device in the sense that the passenger can still be allowed to access the infotainment controls to aide the driver in



(a) (b)

Figure V.8: VUD translation-invariance evaluation. (a) This figure shows the image region-of-interest used to determine the user. The larger rectangle marks the boundary of all the translations of the image patch in measuring translation-invariance of the VUD system. (b) The corresponding average-classification rate for each translation of the image patch is shown in the heat-image to the right. The image and the graph can be used as reference in positioning the camera and image patch location during the installation of the VUD system.

navigational and convenience needs.

V.7 Future Work

For consideration in future work, two aspects of the system can be investigated further.

V.7.A Data collection of larger demographic.

Although the process of data collection was considered at length to ensure a representative training sample, there are other variations that should also have been considered. Namely, all the test subjects were adults and no children were asked to be subjects, and there should be to ensure children are successfully detected. Although one subject had very long sleeves, covering half of his hand, gloves were not used in the data capturing process. The system also does not consider different backgrounds besides that of the vehicle. For this system to function for other vehicles, a training set for that particular vehicle would need to be acquired, something that can be done relatively easily as part of the development of this system for deployment. Although the performance is not expected to decrease by much with these variations, having a representative data-set is critical to ensure that the measured performance is the performance of a deployed system.

V.7.B Upgrading to Affine-Invariance.

Also, to increase the affine-invariance of the image descriptors, the image ROI can be repositioned (re-calibrated) when the controls are visible on a regular basis to correct for any vibrations of the camera over time. A scheme as simple as template matching of the gradient images with the stored image may be used to align the ROI to the location that produces the best classification rates.

VI

Conclusion

This chapter presents a summary of the work in this thesis. We refer the reader to the end of each chapter for a discussion of future research directions.

VI.1 Summary

In this thesis, our motivations were to enable an intelligent system with the knowledge of human desires and wants by developing the necessary concepts, algorithms, and systems to automatically recover human pose and gesture information. We focused on improving techniques for recovering pose and gesture with special emphasis on applications for improving the safety and comfort of vehicles. Our contributions have been in the following areas:

1. Articulated Body Pose Estimation
2. Driver Intersection-turn Intent Recognition
3. In-Vehicle Hand Tracking using Thermal Infrared Imagery.
4. In-Vehicle Vision-based User Determination using Hand images for Infotainment Control Safety

We first presented a novel method for learning and tracking the pose of an articulated body by observing only its volumetric reconstruction. We propose a probabilistic technique that utilizes an extended Gaussian mixture model to describe the spatial distribution of voxels in a voxel image. Each component of the GMM describes a segment or body part, and the collection of components are kinematically constrained according to a pre-specified skeletal model. This model we refer to as a kinematically constrained Gaussian mixture model (kc-gmm), where pairs of components connected at a common joint are encouraged to assume a particular spatial configuration, forming a joint with 1, 2 or 3 degrees-of-freedom (DOF). The pose learning algorithm, based on the EM algorithm, is evaluated using synthesized

hand data, and the HumanEvaII data-set for facilitating comparison among different algorithms. Both data-sets contain ground-truth information for accuracy measurements. A hand model with 16 components, 27 DOF's and a body model with 11 components, 19 DOF's were evaluated. The results show that utilizing volume data, aided only by the degrees-of-freedom constraints, accuracies of joint location estimates within 0.5 cm mean-absolute-error from ground-truth can be achieved with the hand data set and 17 cm MAE from ground-truth can be achieved from subjects S2 and S4 in the HumanEvaII data-set.

Next we present results on the characterization and recognition of driver intent. We focused on the intersection-turn maneuver, but the concepts may apply towards the study of other driving maneuvers. The data-driven approach makes use of vehicle dynamics information and driver head and hand pose information via an optical motion capture system. We describe the details of the preprocessing that resulted in the best performance, and the consideration of the kernel Relevance Vector Machine as the pattern classifier. A series of different measures to examine the performance were proposed, based on true- and false-positive rates and time-aligning the soft response of the classifiers to the start of the maneuver. With this system, we examine the effectiveness of body pose cue for driver intersection-turn intent prediction. Different intent classifiers were trained using vehicle dynamics alone, driver body pose alone, the two together, and other finer permutations of the input cues. We were able to determine that the use of 3-D driver body part position information, in the current state of research, provides only minor benefit beyond recognizing intent with derived gestural cues or vehicle dynamics alone. This is an encouraging finding from the point of view of driver assistance systems development since body pose is a challenging attribute to measure at a distance without the use of entangling markers.

The use of the driver-intent recognition algorithm assumes the availability of body part position information. To fill this requirement, we presented next an in-vehicle system for tracking the position of hands. The first system utilizes a particularly interesting modality, the long-wavelength infrared camera, to detect the movement of the driver's hands. Because the appearance of the driver's hands is more stable in thermal images, a cascade of boosted classifiers using Haar-wavelet like features could be used to detect the hands in images. The detections were used as measurements in a multi-target tracking algorithm based on the Kalman filter and Probabilistic Data Association filter. The results were shown to effectively track the hands over a course of 90 minutes of driving. The results of the hand tracking were combined with steering information to determine any of 5 hand activities over the steering wheel, including grasping the wheel and turning it.

Finally, we presented an in-vehicle system for determining which occupant is accessing the vehicle controls. A vehicle possessing this information can selectively allow access to the infotainment system of the vehicle to the occupant who presents the least danger to the driving. This system utilizes an image patch from a visible-spectrum gray-scale image, and computes a histogram-of-gradients feature to be used in a support-vector-machine classifier. The output of this system represents the 3 possible

outcomes: the driver, passenger or no-one is accessing the infotainment controls. The results are very promising, with detection rates of 97.8% average correct classification rate over 60 minutes of video at 30fps under a variety of moving vehicle operating conditions, including different subjects and lighting conditions.

A

Dynamic Active Display

In this appendix chapter, we introduce the Dynamic Active Display. It is a unique heads-up-display used to present safety-critical visual information to the driver in the driver's view, minimizing deviation of the driver's gaze direction while driving.

A.1 Introduction

Driving an automobile safely depends on events and interactions of three main components of a system: (1) the environment, (2) the vehicle, and (3) the driver (EVD). Human centric computing environments with integrated sensing, processing, networking and displays provide an appropriate framework to develop effective driver assistance systems. Embedded sensors, processors and interfaces have great promise in improving the safety and overall driving experience. However, it is very important that such technologies be introduced in a very careful manner without adverse impact on the attentive state of the driver and safety in traffic. Design of effective "Driver-Vehicle Interfaces" is a multidisciplinary endeavor where expertise in various fields, such as engineering, computer science, cognitive science and psychology is required. Systematic efforts to understand and characterize driver behavior and ethnography surrounding the task of driving are essential in the development of human-centric driver assistance systems.

The key system introduced in this appendix chapter is that of Dynamic active displays (DAD). The DAD presents visual information to the driver in such a way that the driving view and safety-critical visual icons are presented together to the driver, thus minimizing deviation of the driver's gaze direction without adding to unnecessary visual clutter. Dynamic Active Displays have a clear promise to make a positive impact on safety by providing the driver with timely warnings which will reduce the risks associated with accidents due to lane departure, rear-end collisions, vehicles/pedestrians in the blind spots,

limited visibility (night/fog) conditions, providing assistance to the driver in these situations. In this chapter, we describe a framework for holistic sensing and displays for enhancing automobile safety

The new contribution towards the design of the human-centric Intelligent Driving Support Systems is the augmentation with a unique windshield display. Where previously our emphasis was in the fine-grain study of driving behavior, and systems development for extracting EVD information by visual, aural or other modalities, we now close the loop and provide feedback to the driver by way of a novel visual windshield display. In utilizing this display, safety-critical warnings are projected directly into the driver's field-of-view (FOV). The location and timing of these projections are determined by the driver's body posture, awareness and intent. Displaying graphics in a HUD minimizes the time to look away from the road, as well as reduces the need for re-accommodation of focus (an issue of increasing importance for drivers as they age). Studies have shown that time savings of 0.8s to 1.0s in driver reaction time can be achieved with the use of HUDs over conventional heads-down-displays when displaying vehicle warning information [55].

We highlight that by integrating cues associated with the vehicle dynamics, environment, and driver's body pose, these warnings will be provided to allow the driver to take corrective maneuvers only when it is appropriate and at the most appropriate location in the FOV in a manner that is modulated by the criticality/severity index. Our proposed DAD framework is divided into the following 3 subsystems: 1) the Windshield Heads-up-display, 2) Multi-functional Integrated Vision and Sensory Array, and 3) Driver Intent Analysis and Situational Awareness System. The display subsystem presents appropriate information obtained from the other subsystems onto an augmented reality-based heads-up-display, registering it with real objects in the scene using the knowledge of 3D location of the object and eyes, and gaze direction obtained from the sensory array. The multi-functional sensory array collects and analyzes data from video and other sensors to obtain other cues about the driving situation, such as lane position and obstacle positions. Finally, the driver intent analysis and situational awareness system analyzes the visual and other cues from the visual/sensory system to obtain information about driver state, driver situational awareness, and situation criticality to be used to determine what, where, and when the information is presented by the display system.

In the next section, we describe the three (3) display modes of the DAD, and the role of head pose and hand position in displaying alerts when and where appropriate.

A.2 Methods of Display with EVD Cues

Different types of HUDs are reported in the literature [56, 57]. The particular heads-up-display considered in this paper provides a large field-of-view (FOV) display with programmable content drawn using a laser projector. The two prominent features of the display we make use of are 1) large FOV

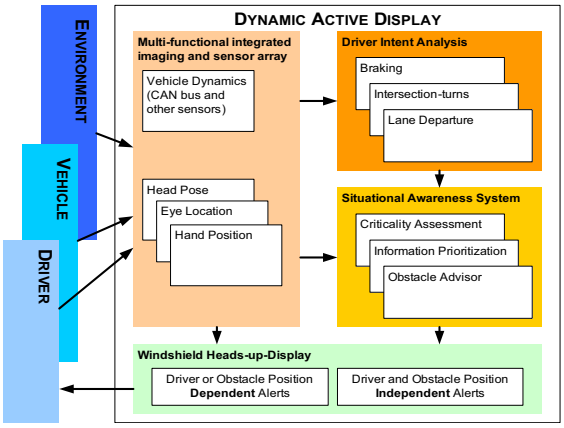


Figure A.1: Data, Information and processing flow in Dynamic Active Display System.

spanning essentially the entire windshield (approximately 65 to 100 degrees in the test vehicle, fig A.2) and 2) high transparency of the display surface. The large FOV attribute is one of the unique aspects of this display. These attributes present several new possibilities in its utility for when the car is in motion. Besides the already established benefits of a HUD in increased eyes-on-the-road time and reduced re-accommodation time, we expect that this type of display has the potential to further increase driver awareness of the driving situation with the system given additional knowledge of driver head pose and hand position. For experimental purposes, a transparent acrylic screen made specifically for projectors is positioned over the windshield with a second non-transparent projection screen placed ahead of the vehicle displaying a video of a forward view of a drive through city streets and freeways to simulate the view while driving. The test set-up is shown in fig. A.3a.

An area that the driver is usually least informed about is the blind spot. On one level, a graphic such as the one shown in fig. A.3b could be displayed to indicate the presence as well as the proximity and dynamics of an obstacle in the driver's blind spot, all without losing peripheral sight of the road ahead. Furthermore, with the knowledge of head pose, the driver can be made aware of potential dangers by way of a visual arrow directing the driver to look at certain directions, as shown in fig. A.3c. Last but not least, with the additional knowledge of hand position and vehicle state, the vehicle can anticipate maneuvers and place emphasis on providing information specific to the maneuver. The vehicle can even refrain from displaying alerts if the driver was already aware of the obstacles ahead. The impact of this feature is foreseeably greatest during lane-change and intersection turn maneuvers, during which time the driver needs to pay attention both to what is in front as well as obstacles on the side of the vehicle.

These scenarios suggest that graphics in a dynamic active display can be displayed in three ways, or display modes: 1) a constant single location on the windshield, 2) at or near the driver's line-of-sight, and 3) on top of obstacles as perceived by the driver just beyond the windshield. For the second and third modes of display, additional knowledge of the driver and environment is required, but it pro-

vides level of danger localization in the automobile that previously could only be achieved in fighter airplanes. Displaying alerts in a constant location is trivial. In the remainder of this section, we describe the procedure to implement functionalities 2) and 3).

The second mode of display, to display a graphic on the windshield along the driver's line of sight, a description of the windshield's surface geometry and the driver's gaze is required. Gaze can either be eye gaze or head gaze for fine or coarse alerts placement. We represent gaze as the gaze origin and gaze normal vector $(\mathbf{o}, \mathbf{d})$, where $\mathbf{o} = (x, y, z)^\top$ and $\mathbf{d} = (dx, dy, dz)$ where $\|\mathbf{d}\| = 1$.This represents 5 degrees-of-freedom in all. In the same coordinate space as the head, the geometry of the display surface can be described as

$$w(\mathbf{p}) = 0 \tag{A.1}$$

where $\mathbf{p} = (x, y, z)^\top$ is a point in $\mathbb{R}^3$ space. Any point that satisfies this equation lies on the windshield in this coordinate space. Finally, points on the windshield in 3D coordinates are transformed to display coordinates by a transformation function $f : \mathbb{R}^3 \mapsto \mathbb{R}^2$

$$\mathbf{u} = f(\mathbf{p}) \tag{A.2}$$

This function can be constructed by assuming a distortion model from a perfect plane, or a table look-up and piece-wise linear interpolation relating windshield points to display coordinates.

The task of displaying an alert just below the driver's line-of-sight for example is then the problem of finding the point $\mathbf{p}^*$ on the line formed by the gaze that intersects the windshield, i.e. finding a point that satisfies equ. A.1. Then to find the corresponding point in display coordinates u^*. The resulting point is transformed using equ. A.2.

We assume for now that the windshield is perfectly planar with a surface normal $\mathbf{n}$ and origin $\mathbf{w}_o$. The windshield surface equation can then be given by

$$w_{\text{planar}}(p) = \begin{bmatrix} \mathbf{n} \\ -\mathbf{n}^\top w_o \end{bmatrix}^\top \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} = 0 \tag{A.3}$$

The points along the line formed by the driver's gaze is

$$\mathbf{p} = \mathbf{o} + \mu\mathbf{d} \quad \forall \mu \in \mathbb{R} \tag{A.4}$$

where μ describes the location of point p as a proportion of $\|\mathbf{d}\|$ along the $\mathbf{d}$ direction from $\mathbf{o}$. The point $\mathbf{p}^*$ that lies on the windshield plane can be found by combining equ. A.3 and A.4, arriving at

$$\mathbf{p}^* = \mathbf{o} + \left(\frac{\mathbf{n}^\top \mathbf{w}_o - \mathbf{n}^\top \mathbf{o}}{\mathbf{n}^\top \mathbf{d}} \right) \mathbf{d} \tag{A.5}$$

Because of our planar assumption, the transformation function f is given by a matrix transformation

$$\mathbf{u}^* = \mathbf{K}\mathbf{R}\mathbf{p}^* = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} & \mathbf{y} & (\mathbf{x} \times \mathbf{y}) \end{bmatrix} \mathbf{p}^* \tag{A.6}$$

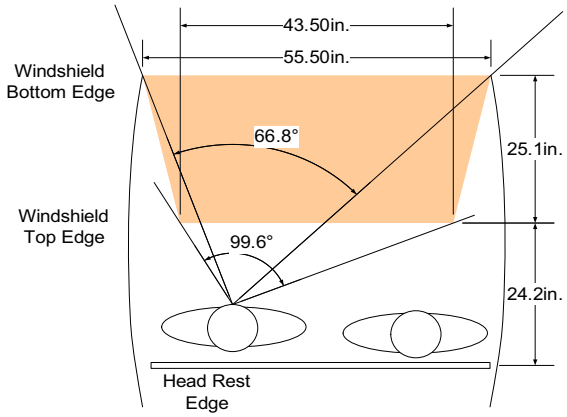


Figure A.2: Diagram illustrates the span occupied by the windshield in the driver's field-of-view.

where $\mathbf{x}$ and $\mathbf{y}$ are the local x- and y-axes of the windshield plane, and a and b are the scaling factors across these axes to translate physical coordinates to display coordinates.

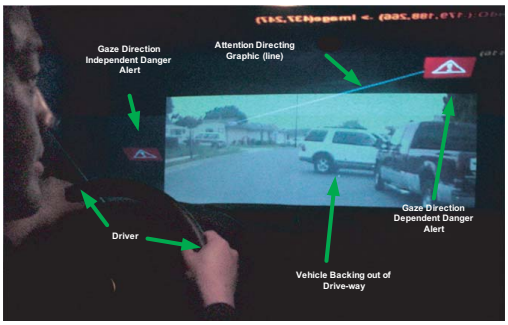
The third mode of display is overlaying alerts on top the driver's view of objects ahead of the vehicle. This requires knowledge of the driver's head position $\mathbf{o}$, the object's position $\mathbf{q}$, the windshield surface geometry $w(\mathbf{p}) = 0$, and windshield transformation function f. The intersection point $\mathbf{p}^+$ between the windshield surface and the line formed between $\mathbf{o}$ and $\mathbf{q}$ is the point on the windshield (in physical coordinates) that lies directly over the driver's view of the object. The transformed point $\mathbf{u}^+ = f(\mathbf{p}^+)$ is the corresponding point in display coordinates.

To demonstrate the efficacy of these display modes, we employed the simulation set-up described above. Head pose is acquired from optical motion capture is used to generate gaze origin and gaze direction. The position of objects ahead of the vehicle is found by annotating the video of object locations in image frames and determining the corresponding location on the forward projection screen, also found through the use of the motion capture system. Fig. A.3b-c illustrates the results of the three display modes: static, gaze aligned, and driver head/obstacle position aligned graphics.

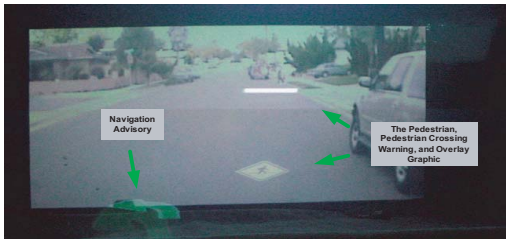
The text of Appendix A, in part, is a reprint of the material as it appears in: Mohan M. Trivedi, Shinko Y. Cheng, "Holistic Sensing and Active Displays for Intelligent Driver Support Systems" IEEE Computer Magazine: Special Issue on Human-Centered Computing, 40(5):60-68, May 2007. I was the primary researcher of the Dynamic Active Display experiments, and the co-author listed in this publication was the primary researcher for the remaining parts, as well as directed and supervised the research which forms the basis of this chapter.

(a) Experimental heads-up-display setup.

(b) Surround monitor.



(c) Directional warnings.



(d) Various warnings.

Figure A.3: These images illustrate (a) the experimental heads-up-display setup used in the paper, (b-d) the various graphics showing the efficacy of gaze/position dependent and independent graphics. Surround awareness graphic. Picture zoomed in to emphasize graphic. Actual size of graphic is approximately 10cm square.

B

LISA-P Test-bed

B.1 Introduction

One of the key criteria in data collection is the capture of representative data of actual operating conditions of the module under development. The need for a vehicular test-bed was clear from an algorithms development standpoint. Data collected from a vehicle in motion will provide data most closely resembling that of the actual data collected from a deployed module.

The research involves the investigation of ways to utilize output from other modules that are also being developed as part of the same research. Algorithms that extract body pose and algorithms utilizing body pose information for activity analysis are simultaneously under investigation. For that reason, we equipped the test-bed with an optical motion capture system while developing an alternative system that performs the same function. This device provides clean, relatively reliable body pose information utilizing markers placed on the body. At the same time, optical mocap also provides ground-truth and training data for data-driven approaches to the body pose estimation problem.

Other considerations of the test-bed pertained to ease of development via accessibility to AC power and data synchronization via time-stamping. The AC power system allows for operation of more prevalent AC-powered equipment in and out of the vehicle, facilitating the preparation of experiments. Time-stamping input allows system designers to consider data as an ensemble and use the data together to interpret the driving situation.

The input cues considered are listed in tab. B.2. In addition to the sensors is a suite of interface cards and devices used to access data from each sensor. They are listed in tab. B.1.

Table B.1: Sensor Interfaces

Device	Description	Source
Euresys Picolo Diligent	NTSC Video PCI capture card	`http://www.euresys.com`
Cygnal C8051F040-TB	CAN Interface Development Board	`http://www.silabs.com`
Keyspan USA-19HS	USB to RS232 Adapter	`http://www.keyspan.com`
Linksys USBBTT100	USB to Bluetooth Adaptor	`http://www.linksys.com`
Sensormatic ROBOT MV87	4-way Video Combiner	`http://www.cctvpros.com`

Table B.2: Sensors installed in the LISA-P.

Sensor Type	Device Specifications	Attribute	Units
Vehicle Dynamics Sensors	Built-in VW Passat 2003 sensors (via CAN bus)	Engine RPM	rpm
		Torque	%
		Throttle	%
		Brake Activation	on,off
		Current Gear	P,R,N,D,2,1
		Wheel Speeds	km/h
		Vehicle Speed	km/h
		Left Turn signal	on,off
		Right Turn signal	on,off
		steering angle	degrees
GPS Receiver	Teletype Bluetooth GPS Receiver Blue-tooth Interface WAAS capable Serial output NMEA protocol `http://www.teletype.com`	Vehicle GPS coordinates	(lat.,lon.)
Grayscale Face and Hand Camera	VidereDesign STH-MDCS-VAR FireWire Interface (IEEE1394) 640x480, 30Hz Rolling Shutter `http://www.videredesign.com`	Images of Driver's face and hands	image
Long-wave infrared hand	ThermalEye 2000AS NTSC Interface 160x120 Microbolometer Sensor Plane 45° fov $7-14\mu m$ wave-length `http://www.thermal-eye.com`	Images of driver's head	image
Color Road-ahead Camera	Hitachi KP-D20A NTSC Interface `http://www.hitachikokusai.com`	Images of road ahead	image
Optical Motion Capture	Vicon (6) SVCam,V8i,iQ2.0 `http://www.viconpeak.com`	3-DOF body part position	mm
		3-DOF body part orientation	radians

B.2 Capture Framework

The information flow diagram is depicted in fig. B.1. This illustration shows the suite of sensors, corresponding interface devices and its connection to the data collection software. At the bottom of the diagram shows the Display Controller via the RGL SDK is the component that controls the graphics on the large-area windshield display. Data capture is performed with up to four independently running processes: LabVIEW, Euresys Capture, Mocap Capture, and FireWire Video Capture. An overview illustration showing the arrangement of the sensors, computers and power systems in the vehicle is shown in fig. B.2. The various sensors and interface devices are illustrated in fig. B.3 and B.4.

A Pentium D processor PC was used as the central capturing and processing computer. The PC is connected to all the devices via a series of interface cards or ports to access the information from the sensors listed in tab. B.2. The primary method of collecting Vehicular Dynamics and GPS data is with National Instruments LabVIEW with its built-in serial bus and file saving functions. Vehicle Dynamics were captured with an additional CAN interface card to translate the CAN packets into an RS232 byte stream. The primary method for collecting optical motion capture data is via TCP/IP network, accessed using the included Tarsus Client SDK software library to communicate with the motion capture data-station. Finally, both infrared and visible wavelength images is captured using an extensible software library to access the NTSC cameras via the Euresys Picolo Diligent video capture boards. The video and body pose information is captured in the same process.

The vehicle dynamics and GPS information is captured using LabVIEW. The script that collects this data is created with a graphical programming language. In fig. B.5, the LabVIEW script is shown. The panel views showing the progress of the capture during deployment can be seen in fig. B.6. The script consists of 3 end-less while loops processing data from 2 serial ports and 1 USB port. The script is started and stopped with a button. Each piece of information is stored in a comma-separated-values file with the hour, minute, second and millisecond when the packet was received.

Both the LabVIEW and custom application accesses the PC's internal millisecond-accurate clock. Upon receiving a packet of information, each piece of software also independently stamps that packet with the current time.

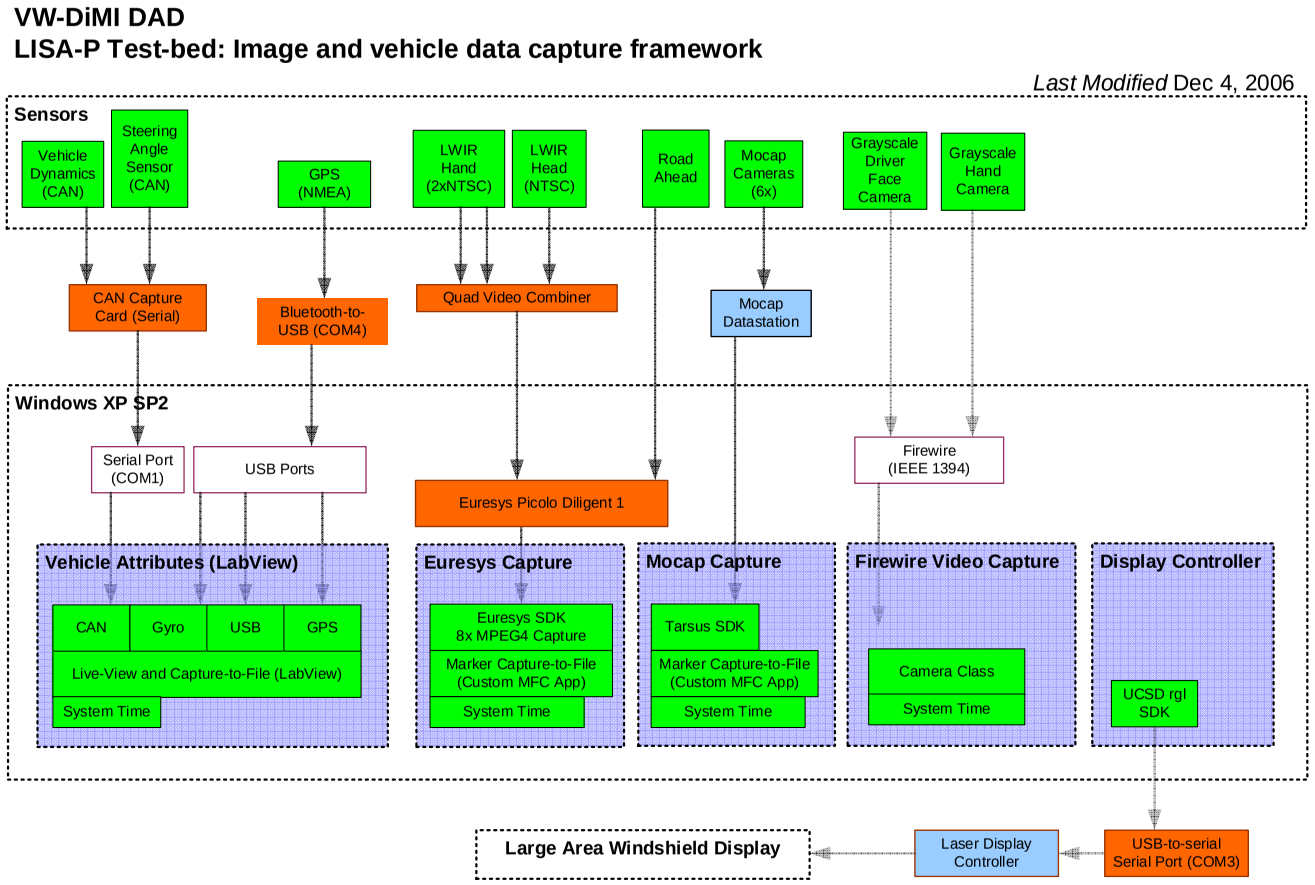


Figure B.1: The LISA-P is equipped with the illustrated sensors, processing software modules and large-area windshield display. Other modules pertain to the type of interface between the sensors and data collection computer, including the optical motion capture data-station.

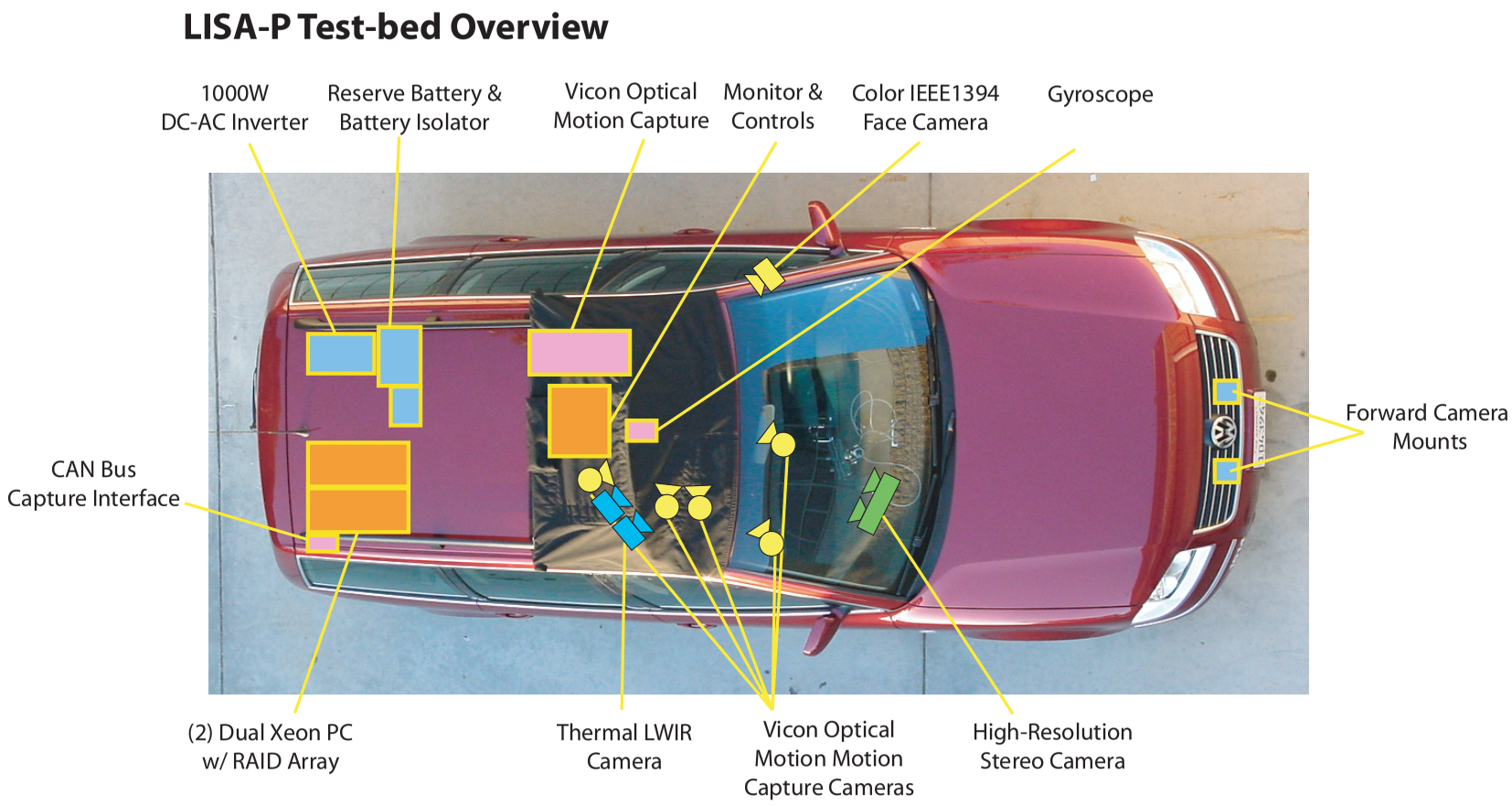
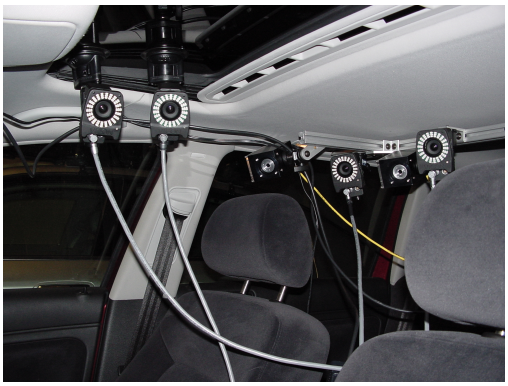
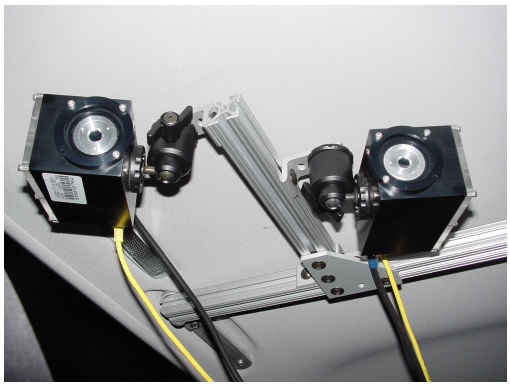


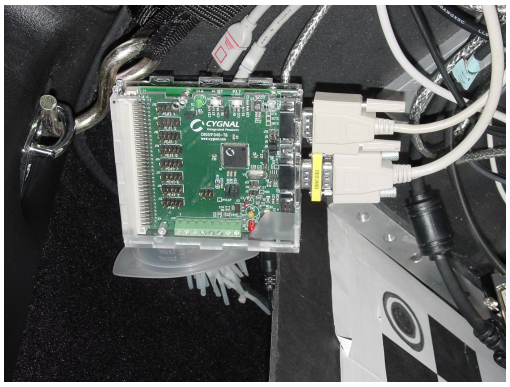
Figure B.2: Overview of the LISA-P and equipment.



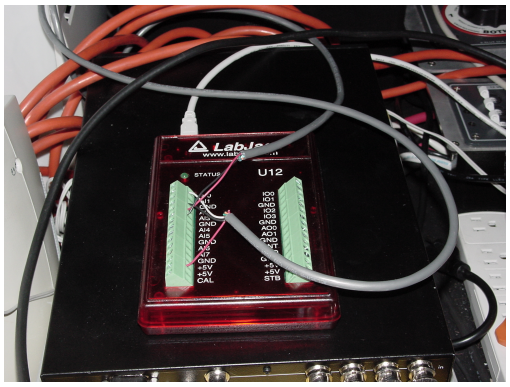
(a) Optical Motion Capture Cameras



(b) ThermalEye Microbolometer Infrared Camera



(c) Cygnal C8051F040-TB CAN Capture Card



(d) U12Labjack USB-based A/D Converter

Figure B.3: Data capture devices provide 6-DOF driver body part pose, thermal imagery of the driver's hands, vehicle dynamics and battery power level.

(a) Blue-tooth GPS Receiver

(b) Hitachi KP-D20A Color CCD Camera

(c) VidereDesign STH-MDCS-VAR gray-scale CMOS Camera

(d) Linksys USBBT100 Blue-tooth Adapter

Figure B.4: Additional data capture devices provide GPS coordinates of the vehicle and images of the head and hands.

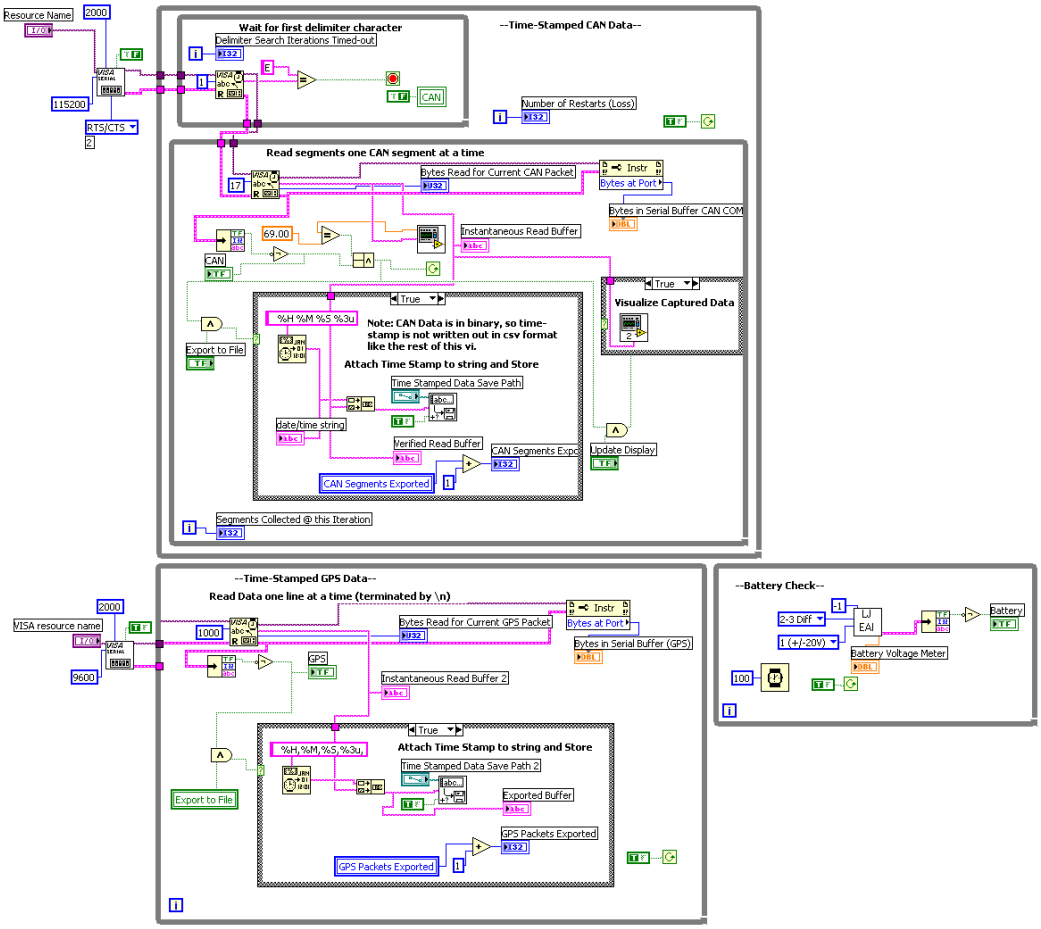
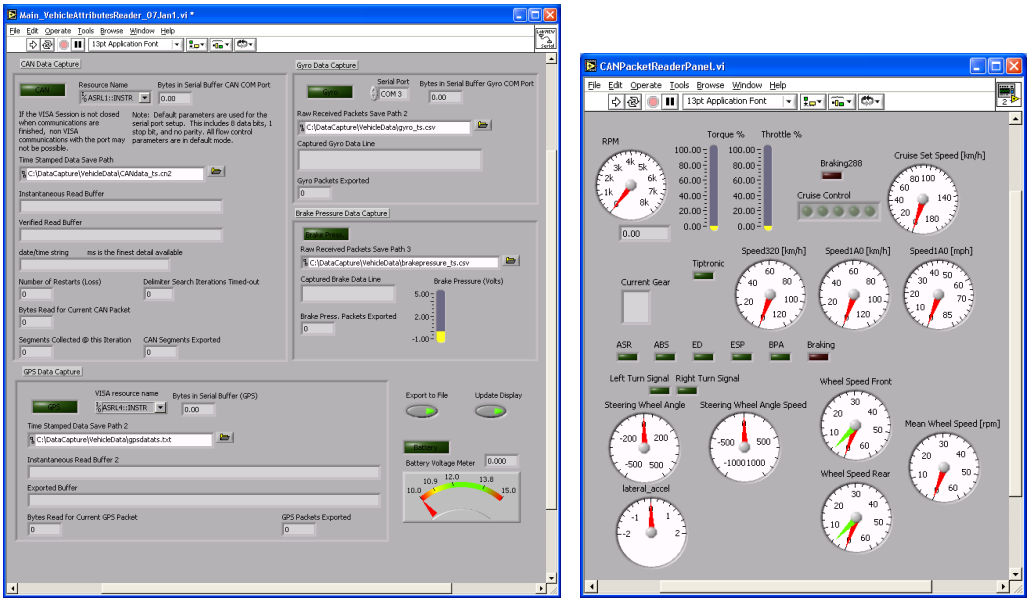


Figure B.5: LabVIEW capture script - Diagram View



(a) Panel view 1 (b) Panel view 2

Figure B.6: LabVIEW capture script - Panel Views.

C

Aligning Two World Coordinate Frames

This section describes the task of relating the world coordinate systems of two rigid body pose estimation systems using the pose estimates of a common body. This algorithm specifically has the Vicon Motion Capture and FaceLab eye gaze and head pose estimation systems in mind, but the algorithm applies to any systems that produce pose estimates of a common rigid body using two separate world coordinate frames.

Suppose System A produces a pose estimate of a head $\{{}^{V}_{A}\mathbf{R}, {}^{V}\mathbf{O}_{A}\}$ and System B produces pose estimates of the same head but with a different coordinate system $\{{}^{F}_{B}\mathbf{R}, {}^{F}\mathbf{O}_{B}\}$. The rotation ${}^{V}_{A}\mathbf{R}$ rotates points from reference frame A (local coordinate frame A) to V (world coordinate frame V) and the translation ${}^{V}\mathbf{O}_{A}$ aligns the origins of the two coordinate frames after rotation. The translation ${}^{V}O_{A}$ is also the location of the local coordinate frame origin in V world coordinates.

To find the coordinates of a point in world coordinates from its coordinates in the local coordinate frame for the Vicon and FaceLab systems, the relationship is given by

$$ {}^{V}\mathbf{p} = {}^{V}_{A}\mathbf{R}\,{}^{A}\mathbf{p} + {}^{V}\mathbf{O}_{A} \tag{C.1} $$

$$ {}^{F}\mathbf{q} = {}^{F}_{B}\mathbf{R}\,{}^{B}\mathbf{q} + {}^{F}\mathbf{O}_{B} \tag{C.2} $$

This information can be verified in the manual for the FaceLab and Vicon systems. As a quick proof, the head location is often given by these systems in a 3-element vector. To find the head location in world coordinates, we set the head locations in local coordinates to $(0, 0, 0)$ and find its locations in world coordinates using (C.4) and (C.5), yielding ${}^{F}\mathbf{O}_{B}$ and ${}^{V}\mathbf{O}_{A}$.

Problem: The problem is defined as estimating the rigid transformation $\{\mathbf{R}, \mathbf{t}\}$ that relates the two estimates of the head (or rigid body) in the respective world coordinate frames. This aligns the two

world coordinate frames so that the head pose estimates from both systems can be directly compared for performance evaluation or other purposes.

Let ${}^{A}\mathbf{p}$ and ${}^{B}\mathbf{q}$ represent the points around the head in Vicon and FaceLab *local* (head) coordinates. Points in these local coordinate frames should equal.

$$ {}^{A}\mathbf{p} = {}^{B}\mathbf{q} = \mathbf{p} \tag{C.3} $$

This replaces (C.1) and (C.2) with

$$ {}^{V}\mathbf{p} = {}^{V}_{A}\mathbf{R}\mathbf{p} + {}^{V}\mathbf{O}_{A} \tag{C.4} $$

$$ {}^{F}\mathbf{p} = {}^{F}_{B}\mathbf{R}\mathbf{p} + {}^{F}\mathbf{O}_{B} \tag{C.5} $$

We assume the same points in FaceLab and Vicon *world* coordinates are related by a rigid transformation

$$ {}^{V}\mathbf{p} = \mathbf{R}{}^{F}\mathbf{p} + \mathbf{t} \tag{C.6} $$

where ${}^{V}\mathbf{p}$ and ${}^{F}\mathbf{p}$ are given by (C.4) and (C.5).

Solution: This solution requires the head pose estimates from the Vicon and FaceLab systems taken of a person at the same time. This yields $({}^{V}_{A}\mathbf{R}, {}^{V}\mathbf{O}_{A})$ and $({}^{F}_{B}\mathbf{R}, {}^{F}\mathbf{O}_{B})$. We can then estimate $(\mathbf{R}, \mathbf{t})$ by first estimating R .

To estimate $\mathbf{R}$ we the `ROT_MATRIX_RANGE` algorithm described in [58]. The algorithm estimates $\mathbf{R}$ by estimating the entries in $\mathbf{R}$ directly, then ensure that the resulting matrix is orthonormal.

The first step is to eliminate $\mathbf{t}$ from (C.6) by taking the difference between two arbitrary points in Vicon world coordinates ${}^{V}\mathbf{p}_i$ and ${}^{V}\mathbf{p}_j$.

$$ \begin{aligned} {}^{V}\mathbf{p}_i - {}^{V}\mathbf{p}_j &= \mathbf{R}{}^{F}\mathbf{q}_i + \mathbf{t} - (\mathbf{R}{}^{F}\mathbf{q}_j + \mathbf{t}) \\ {}^{V}\mathbf{p}_i - {}^{V}\mathbf{p}_j &= \mathbf{R}({}^{F}\mathbf{q}_i - {}^{F}\mathbf{q}_j) \\ \mathbf{n}_k &= \mathbf{R}\mathbf{m}_k \end{aligned} \tag{C.7} $$

At least three such vector equations must be formed to create a fully constrained system of equations to estimate $\mathbf{R}$. From our assumption (C.3), the two points $({}^{V}\mathbf{p}_i, {}^{F}\,\mathbf{q}_i)$ represented in their respective world coordinate frames both originate from a common point in local (head) coordinates $\mathbf{p}_i$. Two points $(\mathbf{p}_i, \mathbf{p}_j)$ (C.7) yields one vector equation with $\mathbf{n}_k$ and $\mathbf{m}_k$. Six points in all are needed to create a fully constrained system of equations.

$$ \mathbf{n}_1 = \mathbf{R}\mathbf{m}_1 \tag{C.8} $$

$$ \mathbf{n}_2 = \mathbf{R}\mathbf{m}_2 \tag{C.9} $$

$$ \mathbf{n}_3 = \mathbf{R}\mathbf{m}_3 \tag{C.10} $$

The points are chosen after the head pose estimates are collected. To ensure that an independent system of equations are created, a good procedure to generating $\mathbf{n}_k$ and $\mathbf{m}_k$ is to set $\{\mathbf{p}_i\}_{i=1}^{6} = \{(1,0,0), (-1,0,0), (0,1,0), (0,-1,0), (0,0,1), (0,0,-1)\}$, and generate $\mathbf{m}_k$ and $\mathbf{n}_k$ for $k = \{1,2,3\}$ from $(i,j) = \{1,2\}$, $i = \{3,4\}$, and $i = \{5,6\}$, respectively. This procedure is however not necessary. The only requirement is that the vectors formed are as orthogonal to each other as possible.

To solve for $\mathbf{R}$, we formulate a least squares problem (C.10) where we minimize

$$\hat{\mathbf{R}} = \arg\min_{R} \{\sum_{k=1}^{N} \|\mathbf{n}_k - \mathbf{R}\mathbf{m}_k\|^2\} \tag{C.11}$$

To solve problem (C.11), we introduce a vector, $\mathbf{r} = [r_{11}, r_{12}, r_{13}, r_{21}, ..., r_{33}]$ formed by the entries of the unknown rotation matrix, and write the associated over-constrained system of equations

$$M\mathbf{r} = \mathbf{f} \tag{C.12}$$

where M is a $3N \times 9$ block-diagonal matrix, and $\mathbf{f}$ is a $3N$-element vector. Compute the least-squares solution, $\hat{\mathbf{r}}$ corresponding to the matrix $\mathbf{R}'$. To enforce $\mathbf{R}' \in$ SO(3), we compute its SVD decomposition $\mathbf{R}' = U\Lambda V^\top$, where

$$\Lambda = V \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & r \end{pmatrix} U^\top \tag{C.13}$$

and set r to 1 or -1, whatever is closest to $\det(VU^\top)$. Finally compute $\hat{\mathbf{R}} = U\Lambda V^\top$ using the corrected Λ. The output $\hat{\mathbf{R}}$ is the best least-squares estimate of the rotation matrix.

Estimating for $\hat{\mathbf{t}}$ is then a matter of solving for $\mathbf{t}$ with any point $\mathbf{p}$ that yields non-zero ${}^{V}\mathbf{p}$ and ${}^{F}\mathbf{p}$ in (C.3).

$$\hat{\mathbf{t}} = {}^{V}\mathbf{p} - \hat{\mathbf{R}}\,{}^{F}\mathbf{p} \tag{C.14}$$

$$= {}^{V}_{A}\mathbf{R}\mathbf{p} + {}^{V}\mathbf{O}_A - \hat{\mathbf{R}}({}^{F}_{B}\mathbf{R}\mathbf{p} + {}^{F}\mathbf{O}_B) \tag{C.15}$$

Using this rotation and translation, one can translate the camera position

D

Vision-based User Determination System Development Kit

A set of executables and libraries have been prepared for further development of the VUD module as part of Technology Transfer. The package is titled "UCSD CVRR Hand Identification System Development Kit" The components of this package are listed as follows:

Applications:

AnnotateWidgetStatus This windows console application reads and annotates whose hands are in the widget area as defined by rectangle parameters in this windows application. Certain key strokes can navigate through the video.

hi_create_samples This windows console application takes as input arguments 1) the location of the region-of-interest to extract the HOG features, and 2) the location of the annotation file to prep the associated design and target matrix for classifier training. The program then concatenates them into a single D and T matrix which is then saved as yml and csv files.

kNN_train This windows console application takes the D and T files in yml format and trains a multi-class k-NN classifier. M-fold cross-validation as well as the maximum number of samples of each class to be considered can be configured.

svm_train This windows console application takes the D and T files and trains a multi-class SVM classifier. M-fold cross-validation as well as the maximum number of samples of each class to be considered can be configured.

hi_app This windows console application processes a video file with the hand identification system.

Output files include response (class predication) file, confusion matrix file, and output video file. The required input files are the annotation file, .svm file, input video, and image ROI parameters.

Libraries:

cvrr_core This library contains useful functions for applications above regarding windows console applications, consisting of functions for argument parsing and exporting matrices to csv files.

HandID_Common This library contains functions specifically used for user determination. This includes functions such as reading annotation files, reading D and T yml files. It also contains the TransRect class used for easily iterating over different sizes of rectangles.

ml_common This library contains function specifically used for machine learning. This includes functions such as finding the number of categories, randomly permuting the elements of an array containing indices to the m sample sets for m-fold cross validation, and generating the `sample_idx` indicating samples to be considered for training.

The applications are typically called in the order listed when training from the captured video. A set of samples are also provided to illustrate the usage of these windows console applications.

References

[1] A. Kendon, *Gesture: Visible Action as Utterance*, Cambridge University Press, 2004.

[2] V. I. Pavlovic, R. Sharma, and T. S. Huang, IEEE Trans. on Pattern Analysis and Machine Intelligence **19** (1997) 677.

[3] Traffic Safety Facts, Technical Report DOT HS 810 791, U.S. Department of Transportation, National Highway Traffic Safety Administration, 2007.

[4] B. N. Campbell, J. D. Smith, and W. G. Najm, Examination of Crash Contributing Factors Using National Crash Databases, Technical Report DOT HS 809 664, U.S. Department of Transportation, National Highway Traffic Safety Administration, 2003.

[5] T. A. Dingus *et al.*, The 100-Car Naturalistic Driving Study: Phase II - Results of the 100-Car Field Experiment, Technical Report DOT HS 810 593, U.S. Department of Transportation, National Highway Traffic Safety Administration, 2006.

[6] G. Slabaugh, B. Culbertson, and T. Malzbender, A survey of methods for volumetric scene reconstruction for photographs, in *International Workshop on Volume Graphics*, pages 81–100, 2001.

[7] K. N. Kutulakos and S. M. Seitz, Intl. J. Computer Vision **38** (2002) 199.

[8] A. Laurentini, Computer Vision and Image Understanding **67** (1997) 81.

[9] G. Cheung, S. Baker, and T. Kanade, Shape-from-Silhouette of Articulated Objects and its Use for Human Body Kinematics Estimation and Motion Capture, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, pages 77–84, 2003.

[10] T. B. Moeslund, A. Hilton, and V. Kruger, Computer Vision and Image Understanding, Special Issue on Modeling People: Vision-based understanding of a persons shape, appearance, movement and behaviour **104** (2006) 90, doi:10.1016/j.cviu.2006.08.002.

[11] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, A Review on Vision-Based Full DOF Hand Motion Estimation, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 75–82, 2005.

[12] D. Gavrila, Computer Vision and Image Understanding **73** (1999).

[13] G. K. M. Cheung and T. Kanade, A Real-Time System for Robust 3D Voxel Reconstruction of Human motions, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 714–720, 2000.

[14] I. Mikic, M. M. Trivedi, E. Hunter, and P. Cosman, Intl. J. Computer Vision **53** (2003) 199.

[15] E. A. Hunter, P. H. Kelly, and R. C. Jain, Estimation of Articulated Motion Using Kinematically Constrained Mixture Densities, in *IEEE Proc. Nonrigid and Articulated Motion Workshop*, pages 10–17, 1997.

[16] S. Y. Cheng and M. M. Trivedi, Multimodal Voxelization and Kinematically Constrained Gaussian Mixture Model for Full Hand Pose Estimation: An Integrated Systems Approach, in *IEEE Proc. Int. Conference on Computer Vision Systems*, pages 34–42, 2006.

[17] E. Ueda, Y. Matsumoto, M. Imai, and T. Ogasawara, IEEE Transactions on Industrial Electronics **50** (2003) 676.

[18] K. Ogawara, K. Hashimoto, J. Takamatsu, and K. Ikeuchi, Grasp Recognition using a 3D Articulated Model and Infrared Images, in *IEEE/RSJ Proceedings of Conference on Intelligent Robots and Systems*, volume 2, pages 27–31, 2003.

[19] C. Theobalt, E. de Aguiar, M. Magnor, H. Theisel, and H.-P. Seidel, Marker-free Kinematic Skeleton Estimation from Sequences of Volume Data, in *ACM Symposium on Virtual Reality Software and Technology*, pages 57–64, 2004.

[20] G. J. Browstow, I. Essa, D. Steedly, and V. Kwatra, Novel Skeletal Representation for Articulated Creatures, in *Proc. European Conf. on Computer Vision*, volume 3, pages 66–78, 2004.

[21] G. K. M. Cheung, *Visual Hull Construction, Alignment and Refinement for Human Kinematic Modeling, Motion Tracking and Rendering*, PhD thesis, Robotics Institute, Carnegie Mellon University, 2003, Technical Report CMU-RI-TR-03-44.

[22] J. Yan and M. Pollefeys, Automatic Kinematic Chain Building from Feature Trajectories of Articulated Objects, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 712–719, 2006.

[23] A. P. Dempster, N. M. Laird, and D. B. Rubin, Journal of the Royal Statistical Society **39** (1977).

[24] E. Hunter, *Visual Estimation of Articulated Motion using Expectation-Constrained Maximization Algorithm*, PhD thesis, University of California, San Diego, 1999.

[25] S. Y. Cheng and M. M. Trivedi, Multimodal Voxelization and Kinematically Constrained Gaussian Mixture Model for Full Hand Pose Estimation: An Integrated Systems Approach, in *IEEE International Conference on Computer Vision Systems*, 2006.

[26] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer Science+Business Media, LLC, 2006.

[27] L. Sigal and M. J. Black, HumanEva: Synchronized Video and Motion Capture Dataset for Evaluation of Articulated Human Motion, Technical Report CS-06-08, Department of Computer Science, Brown University, Providence, Rhode Island 02912, 2006.

[28] H. Attias, Inferring parameters and structure of latent variable models by variational Bayes, in *Conference on Uncertainty in Artificial Intelligence*, 1999.

[29] P. Merrell *et al.*, Real-time visibility-based fusion of depth maps, in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[30] D. Salvucci, H. M. Mandalia, and N. K. T. Yamamura, Human Factors **39** (2007) 532.

[31] D. Salvucci, Inferring driver intent: A case study in lane-change detection, in *Human Factors Ergonomics Society*, 2004.

[32] N. Oliver and A. Pentland, Graphical Models for Driver Behavior Recognition in a SmartCar, in *IEEE Proceedings on Intelligent Vehicles Symposium*, pages 7–12, 2000.

[33] A. Pentland and A. Liu, Neural Computation **11** (1999) 229.

[34] N. Kuge, T. Yamamura, O. Shimoyama, and A. Liu, Society of Automotive Engineers Transactions **2000-01-0349** (1998).

[35] W. G. Najm, J. D. Smith, and D. L. Smith, Analysis of Crossing Path Crashes, Technical Report DOT HS 809 423, U.S. Department of Transportation, National Highway Traffic Safety Administration, 2001.

[36] J. C. McCall and M. M. Trivedi, Trans. on Intelligent Transportation Systems **8** (2007).

[37] S. Y. Cheng, S. Park, and M. M. Trivedi, Computer Vision and Image Understanding (2006), doi: 10.1016/j.cviu.2006.08.010.

[38] C. Bishop, *Pattern Recognition and Machine Learning*, Information Science and Statistics, Springer Sciences and Business Media, LLC, 2006.

[39] P. Viola and M. Jones, Rapid Object Detection using a Boosted Cascade of Simple Features, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2001.

[40] R. Lienhart and J. Maydt, An Extended Set of Haar-like Features for Rapid Object Detection, in *IEEE Intl. Conf. on Image Processing*, 2002.

[41] Y. Bar-Shalom and X.-R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*, Yaakov Bar-Shalom, Massachusetts, 1995.

[42] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, Boston, 1999.

[43] S. Park and J. K. Aggarwal, Semantic-level Understanding of Human Actions and Interactions Using Event Hierarchy, in *Workshop on Articulated and Nonrigid Motion, in conjunction with the Conference on Computer Vision and Patter Recognition*, 2004.

[44] J. D. Lee, Journal of Safety Research **38** (2007) 203.

[45] M. Jerbi, S.-M. Senouci, R. Meraihi, and Y. Ghamri-Doudane, An Improved Vehicular Ad Hoc Routing Protocol for City Environments, in *IEEE Int'l Conf. on Communications*, pages 3972–3979, 2007.

[46] E. Murphy-Chutorian, A. Doshi, and M. M. Trivedi, Head Pose Estimation for Driver Assistance Systems: A Robust Algorithm and Experimental Evaluation, in *IEEE International Conference on Intelligent Transportation Systems 2007*, 2007.

[47] P. B.-L. Chou, J. Lai, A. Levas, and P. A. Moskowitz, United States Patent (2001), 6,181,996.

[48] J. Joseph E. Harter, G. K. Scharenbroch, W. W. Fultz, D. P. Griffin, and G. J. Witt, United States Patent (2003), 6,668,221.

[49] M. Kölsch and M. Turk, Analysis of Rotational Robustness of Hand Detection with Viola & Jones' Method, in *IAPR International Conference on Pattern Recognition*, 2004.

[50] Q. Yuan, S. Sclaroff, and V. Athitsos, Automatic 2D Hand Tracking in Video Sequences, in *IEEE Workshop on Applications of Computer Vision*, pages 250–256, 2005.

[51] C. Burges, Knowledge Discovery and Data Mining **2** (1998).

[52] C.-C. Chang and C.-J. Lin, A Library for Support Vector Machines, website, 2007.

[53] K. Mikolajczyk and C. Schmid, IEEE Transactions on pattern Analysis and Machine Intelligence **27** (2005) 1615.

[54] D. G. Lowe, International Journal of Computer Vision **60** (2004) 91.

[55] Y.-C. Liu and M.-H. Wen, Int. J. of Human-Computer Studies **61** (2004) 679.

[56] K. W. Gish and L. Staplin, Human Factors Aspects of Using Head Up Displays in Automobiles: A Review of the Literature, Technical Report DOT HS 808 320, U.S. Department of Transportation, National Highway Traffic Safety Administration, 1995.

[57] H. Watanabe, H. Yoo, O. Tsimhoni, and P. Green, The Effect of HUD Warning Location on Driver Responses, in *International Transportation Systems World Congress*, pages 1–10, 1999.

[58] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*, Prentice Hall, 1998.