Efficient Hand Orientation and Pose Estimation For Uncalibrated Cameras

Muhammad Asad

Department of Computer Science

City, University of London

A thesis submitted in fulfillment of the requirement for the degree of

Doctor of Philosophy

City, University of London August 2017

In loving memory of my grandfather, Rehmat Ali.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration, except where specifically indicated in the text. This dissertation contains less than 65,000 words including appendices, bibliography, footnotes, tables and equations and has less than 150 figures. I grant powers of discretion to the City, University of London librarian to allow the dissertation to be copied in whole or in part without further reference to myself (the author). This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgment.

Muhammad Asad

August 2017

Acknowledgements

First of all, I praise Allah Almighty, who taught me, inspired me and guided me throughout my life and, in particular, during my time at City. I am humbled by the strength He gave me and for His countless blessings that have helped me through difficult times.

I would like to extend my sincerest gratitude to my supervisor Greg Slabaugh, for giving me the opportunity to pursue my research, and for his guidance, encouragement, and advice throughout my time as his student. I have been extremely blessed to have gained knowledge from a supervisor who cared so much about my work, and who always had a constructive response to my questions. I would also like to thank my second supervisors Panos Liatsis and Artur Garcez, for their valuable discussions and constructive comments, that are greatly reflected in my research.

I have been fortunate enough to have worked alongside excellent researchers, in particular I would like to extend special thanks to the Computer Vision Group (CVG) at City. I am grateful for my fellow researchers, who have transformed CVG from just a couple of members to a fully grown research group with diverse ideas. I owe special thanks to S. M. Masudur Rahman Al-Arif, Atif Riaz, Anfisa Lazareva and Rilwan Remilekun Basaru for the countless discussions and the collaborative research work that we shared. I gratefully acknowledge the valuable feedback on my thesis from my colleagues including Aisha Junejo, Anfisa Lazareva, S. M. Masudur Rahman Al-Arif, Atif Riaz, Nathan Olliverre, Socrates Katsoulacos and Rilwan Remilekun Basaru.

I gratefully acknowledge the Fee Waiver award from City, that allowed me to pursue my research degree and the Graduate School Conference Attendance funds that enabled me to present my work in International Conferences. I acknowledge the support of my supervisor

throughout my degree in enabling me to present my work at different venues. I am honored and humbled by the Best Paper Award and Travel Grant Award at CVPR Workshop on Observing and Understanding Hands in Action, 2016.

I would also like to acknowledge the work opportunities provided to me by Learning Enhancement and Development (LEaD) team at City. I have been lucky to work alongside such a talented team of Educational Technologists. I owe special thanks to Sandra Partington, Olivia Fox and Stef Smith for helping me develop throughout my time at LEaD and enabling me to understand and improve my professional skills.

I thank my examiners, Gabriel Brostow and Chris Child, for reviewing and understanding my work, and for their constructive feedback that has helped in further improving this dissertation. I also extend sincere gratitude to the Postgraduate Research Course Officer, David Mallo-Ferrer, who ensured that everything related to my thesis submission and examination went smoothly. I would also like to thanks Evangelia Kalyvianaki, who chaired my Viva examination.

I thank my parents and siblings for their continuous love and support, without which this work would not have been possible.

Abstract

Recent trends in wearable technology have resulted in the introduction of a number of egocentric devices that provide a first-person view of the world [1]. While the technology is evolving rapidly, limited natural interaction methods are proving to be a major hurdle in exploring the usefulness of such devices. The widespread use of existing hand pose estimation methods is restricted by a number of factors that include lack of generalization, usage of depth images and the requirement for calibration and initialization steps [2, 3]. Furthermore, most of these methods do not address hand pose estimation under varying viewpoint, which is a major challenge in first-person vision. Thus, there is a need for a new hand orientation and pose estimation method that can generalize well for a number of variations including orientation, hand pose, shape, size and style.

We propose a unified framework for orientation estimation for a planar hand pose. This framework is used to introduce three novel orientation estimation methods, namely, single-layered single-variate Random Forest (SL-RF SV), multi-layered Random Forest with Marginalization through Regression (ML-RF MtR) and staged probabilistic regression (SPORE). We also present a method for extracting contour-based features from hand silhouette images called Contour Distance Features. The SL-RF SV hand orientation regression method utilizes dimensionally reduced Contour Distance Features to learn the mapping of silhouette images onto hand orientation angles. The ML-RF MtR expands on the ability of SL-RF SV, by dividing the data into subsets and learning expert regressors from each subset. The main contribution of this method is the use of a regressor for learning marginalization of multiple trained expert regressors. We further generalize this method to propose SPORE, which trains expert regressors in stages from multiple variations in a dataset. This method relies on intermediate model evaluations to define harder samples, which are used to train subsequent stages in the model. We also extend SPORE to simultaneously infer hand orientation and pose. We provide comprehensive experimental validation of the proposed methods and compare them to the state-of-the-art for multi-layered Random Forest and hand orientation regression methods. The experimental validation results indicate that SPORE outperforms

existing marginalization as well as hand orientation regression methods with a mean absolute error of 8.42° for Azimuth angle and 7.38° for Elevation angle. Our experimental results also show that the introduction of hand orientation in pose classification enables the model to build a better understanding of hand pose under varying viewpoints as compared to a method that learns from hand pose data only. Furthermore, we present a number of Augmented Reality applications that show the significance of the proposed methods in facilitating novel hand-based interactions.

We also propose a framework for capturing automatic annotation of ground truth hand orientation angles in a dataset. We contribute a painting game application-based method that enables collection of large hand orientation and pose datasets from a number of participants. To this end, we collect four different hand orientation datasets containing a total 26,772 samples from up to 22 participants using up to four different planar hand poses.

Table of contents

List of figures

List of tables

Glossary of abbreviations

CDF Contour Distance Feature(s). xv, 37–41, 43, 44, 57–59, 61, 62, 66, 70, 72, 75–78, 80, 81, 83, 87, 97, 101, 102, 107, 119, 122

CMAE Combined Mean Absolute Error. 52, 54, 106–108

CNN Convolutional Neural Networks. 16–18, 20, 28, 37

CPU Central Processing Unit. 66, 86, 103

Dof Degrees-of-Freedom. 3, 12, 15

GPU Graphics Processing Unit. 3, 14, 15, 26, 28

GT Ground Truth. xv–xvii, 9, 18, 29, 33, 36, 41, 46, 47, 52, 53, 55, 58, 59, 62, 63, 65, 66, 68–70, 77, 79, 81, 82, 84, 85, 88–90, 93, 99–103, 110–116

IMU Inertial Measurement Unit. 24, 126

MAE Mean Absolute Error. xvi, xix, 52, 54, 63–67, 72, 89, 107–109, 113, 115

ML-RF Multi-layered Random Forest. 21, 24, 75, 86, 87, 95, 96, 104, 105, 109, 123

ML-RF MtR Multi-layered Random Forest with Marginalization through Regression. 76–79, 85, 86, 88–90, 92, 94, 95, 104, 105, 108, 110–113, 115, 122, 123, 128, 129

ML-RF MtR K-means Multi-layered Random Forest Marginalization through Regression with K-means Clustering. 104, 108, 112, 113, 115

MLE Maximum Likelihood Estimate. 45, 61, 80, 81, 92, 99

NN Neural Networks. 61, 63–67

PCA Principal Component Analysis. 57, 58

PDF Probability Density Function. 32

PMF Probability Mass Function. 32

PSO Particle Swarm Optimization. 12, 14, 20, 22, 24

RANSAC Random Sample Consensus. 46, 85

RF Random Forest. 16, 17, 24–26, 29, 40–44, 57–59, 61, 77, 80, 83, 87, 88, 97, 101, 105, 106, 108, 109, 115

RF Clf Random Forest Classifier. xvii, xix, 105, 114–116

SL-RF SV Single-layered Single-variate Random Forest. xix, 61–67, 70, 74, 86, 88, 89, 104, 108, 109, 115, 122, 123

SPORE Staged Probabilistic Regression. xiii, xvii, xix, 54, 95–118, 123, 126–129

SVM Support Vector Machines. 42

ToF Time-of-Flight. 14

Chapter 1

Introduction

The ability to manipulate and interact with different objects comes naturally to us. These abilities are mainly driven by our curiosity to acquire more knowledge to improve our everyday life. Today, we are entering a new era of Augmented and Virtual Reality, where the research of more than 30 years is seeing many of the final pieces falling into place [4–6]. Augmented Reality enables real-time presentation of virtual objects in the context of physical spaces surrounding a user [7]. Virtual Reality, on the other hand, enables its users to explore environments that are literally out of this world [8]. However, current devices for both these alternate realities provide users with limited manipulation and interaction abilities, which are essential for us to effectively utilize these devices. To this end, determining hand orientation and pose is critical for understanding the manipulation and interaction with virtual objects in these alternate realities. Thus, an out-of-the-box solution is sought that can estimate hand orientation and pose across different users without the need for camera calibration.

This dissertation is devoted to models that can learn the mapping of hand silhouette images onto the orientation and pose of hands. These proposed models utilize uncalibrated 2D monocular hand images while aiming to generalize across a dataset captured from different users. In the following sections, the background and motivation for this work are described to define the problem addressed in this work. The aims of this dissertation are then laid out, formulating the content of our research. Contributions of the underlying research

are then listed, followed by the list of publications arising from this work. An overview of the content of this dissertation concludes this chapter.

1.1 Background and Motivation

The recent advancements in technology have resulted in the introduction of powerful wearable and mobile computing devices. The key idea behind their widespread success is the human-centric design. These devices focus on adapting themselves to users' lifestyle, in contrast to the existing technology that requires users to conform to the available interfaces, such as keyboards and mice. This shift in focus has also helped promote the use of computers from passive accessories, such as laptops and desktops, to being an integral part of our daily life, like smartphones, smartglasses and smartwatches [9–12]. These devices carry a range of sensors to sample and understand the users' activities, which are then used to enhance the user experience. Furthermore, recent interest in Augmented and Virtual Reality has shown their applicability in a wide range of applications. The true potential of these devices, however, is confined by the tangible or voice-based user interface. For this reason, there has been increased interest from the research community to utilize the available sensors, especially video cameras, for presenting natural interaction methods [13]. Video cameras on such devices capture the scene from a user's perspective which is also known as egocentric perspective. The unique ability of this perspective to capture hand interactions from a first person view motivates the need for understanding and utilizing hand-based interaction in this new setting. **This dissertation explores new ways to detect and utilize hand-based interactions.**

Over the recent years, real-time depth cameras have facilitated the introduction of a range of novel natural interaction methods [14]. These cameras enable acquisition of depth images that encode the visible 3D shapes using distance values of objects in view. The acquired depth images are proving fundamental for addressing the human body and hand pose estimation problems [3, 15, 16]. Furthermore, recent research has focused on addressing hand pose estimation under challenging settings using depth images [17–20].

While depth cameras are demonstrating to be of great significance for addressing the hand pose inference problem, these cameras are not widely available on mobile devices due to the considerations of power consumption, cost and form-factor [21]. In contrast, 2D monocular cameras are readily available in majority of mobile devices. Further, existing hand pose estimation methods have not seen widespread adoption for mobile devices. This is due to limited accuracy and efficiency of the existing methods. Most existing methods do not take into account the inter-person hand shape variations [3]. Moreover, some existing methods also require specialized hardware (e.g. high-end Graphics Processing Unit GPU) which have limited availability on wearable and mobile devices [22]. Some existing methods also require camera calibration which is a laborious process, presenting limitations in a practical context [19, 22–25]. **This dissertation focuses on methods that learn hand orientation and pose using 2D uncalibrated monocular images in new ways. To this end, we present machine learning models that are capable of generalizing well under a range of variations resulting from inter-person hand shape, size and style variations.**

Most existing hand pose estimation methods have diverse datasets that focus on variations in hand pose [3]. However, these datasets are limited in terms of viewpoint or hand orientation, which is an integral source of variation given high Degrees-Of-Freedom (Dof) of the hand [3]. Furthermore, when utilizing 2D monocular images, the hand orientation variations can significantly reduce the accuracy of any hand pose estimation method as changes in hand orientation can result in self-occlusions and significant transformation of the projected hand shape. Thus, hand orientation is the first step towards determining the detailed articulation of the hand. A number of previously proposed model-based hand pose estimation methods utilize 2D monocular images to infer the global hand orientation along with the detailed hand pose [26, 27]. These methods can significantly benefit from a hand orientation inference method that learns the mapping of 2D monocular images to hand orientation angles. Furthermore, when used in Augmented Reality applications, the inferred hand orientation can provide the user direct control of the orientation of augmented objects [28]. We observe that the changing orientation of the hand induces changes in the projected hand shape. In such cases, a model that learns the relationship between features related to hand shape and the

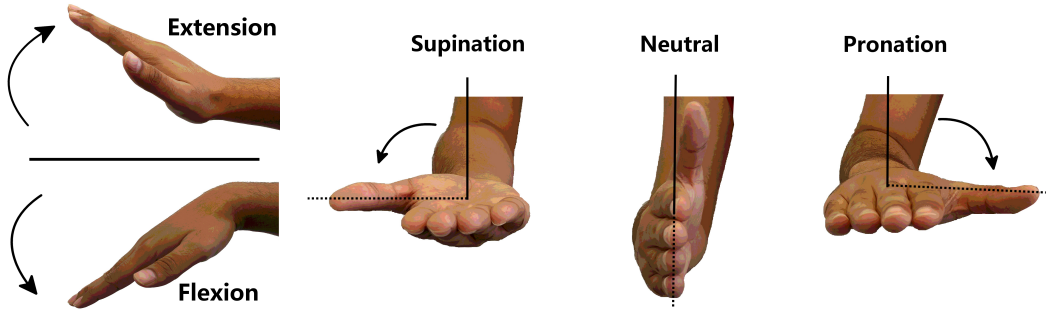


Fig. 1.1 Movements in the wrist and the forearm used for hand orientation regression shows (a) flexion and extension of the wrist and (b) supination and pronation of the forearm.

orientation angles would highly contribute towards understanding and using different hand postures. **In this work, the hand orientation is defined by azimuth and elevation angles that correspond to the flexion and extension of the wrist and supination and pronation of the forearm measured along the azimuth and elevation axis [29] (as shown in Fig. 1.1)**

1.2 Problem Definition

Given a dataset $\mathcal{S} = \{\mathbf{s}_k, \mathbf{o}_k\}_{k=1}^{K}$ of K hand silhouette images $\mathbf{s}_k$ and the corresponding hand orientation $\mathbf{o}_k$ captured using a planar hand pose, such that it captures variation in shape, size, style and orientation from a number of different participants. The problem of hand orientation estimation aims at (i) extracting reliable hand shape features $\{\mathbf{d}_k\}_{k=1}^{K}$ and (ii) training a model to learn the mapping $f : \{\mathbf{d}_k\}_{k=1}^{K} \mapsto \{\mathbf{o}_k\}_{k=1}^{K}$ from $\{\mathbf{d}_k\}_{k=1}^{K}$ to the hand orientation angles $\{\mathbf{o}_k\}_{k=1}^{K}$. The orientation $\mathbf{o}_k$ contains a pair of angles (ϕ_k, ψ_k), where ϕ_k and ψ_k are the azimuth and elevation angles along the two major axes of the hand, namely, the azimuth and elevation rotation axes, respectively. Fig. 1.2 shows the problem formulation for our hand orientation inference methods. We aim to address the problem of generalization of hand orientation and pose from a given dataset without requiring any calibration or initialization and in the presence of a number of inter-person hand variations.

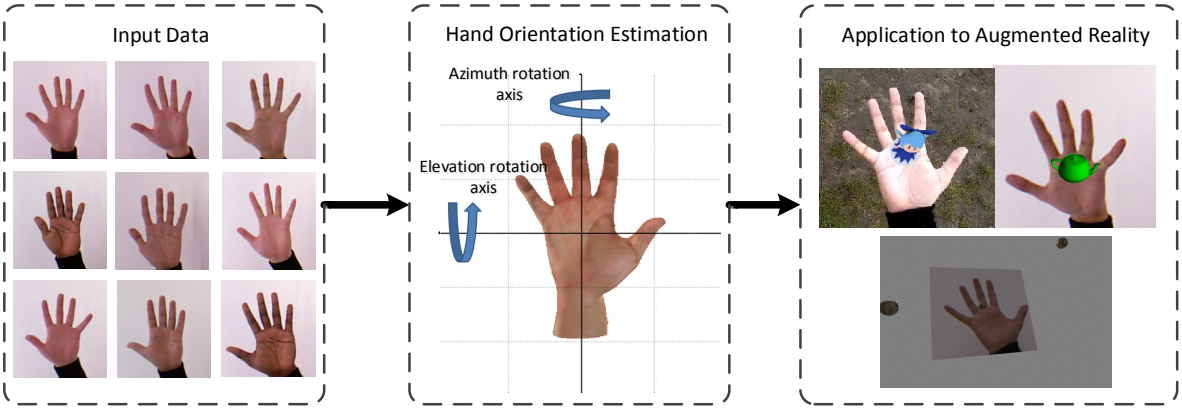


Fig. 1.2 Hand orientation inference problem formulation. For a dataset capturing variations in the hand shape, size, style and orientation, the problem of hand orientation inference learns the mapping of silhouette images onto the corresponding hand orientations. The extracted orientations can be used in a number of applications, including Augmented Reality.

The extracted orientation angles can be used to realize an immersive Augmented Reality application, where orientation manipulations from the hand can facilitate transforming virtual objects. We further note that these orientation angles can also be used to reduce the search space for existing generative hand pose estimation methods that optimize against global and local hand parameters.

1.3 Research Aims

The goal of this dissertation is to contribute and evaluate methods for learning models from hand orientation and pose data. Our aim is to propose frameworks that are able to learn from a number of variations in the data. The limitations of the existing literature facilitate us to put forward the following research questions:

1. Can recently introduced depth cameras be used to acquire a hand orientation and pose dataset that contains 2D monocular images and additional 3D orientation information?

And, if so, can we introduce a feedback system for reliable capture of data? This question is explored in Chapters 3 and 5.

2. Is it possible to use a machine learning model to learn the mapping of hand silhouette to hand orientation? This question is explored in Chapters 4, 5 and 6.

3. Can such model(s) cover the inter-person and intra-person hand shape, size and style variations? And, if so, can we propose efficient and flexible frameworks that can exploit and learn from a number of variations within a dataset? This question is explored in Chapters 5 and 6.

4. Does the additional hand orientation information contribute towards improving the hand pose estimation under varying viewpoints? This question is explored in Chapter 6.

In the next section, we present the original contributions of this dissertation that address the above-mentioned research questions.

1.4 Contributions

The variations in the shape, size and style of the hand across different users makes the hand orientation inference a challenging task. Moreover, the 2D uncalibrated monocular image only captures the projected hand shape, where multiple hand orientations can produce the same captured shape. The main contributions of this dissertation with respect to the problem defined in Section 1.2 and the research questions raised in Section 1.3 are as follows:

1. We propose a framework for learning hand orientation from a dataset of hand silhouette images and orientation angles [28]. We design a contour-based feature extraction method, that enables reliable extraction of hand shape features from a dataset captured from different users. We also contribute a method for automatic hand orientation annotation for planar hand poses. The framework trains a Random Forest regressor to learn the mapping of contour-based features to the corresponding hand orientations.

2. We contribute four datasets of 2D monocular hand images and corresponding orientation, containing a total of 26,772 samples, captured from up to 22 participants and containing up to four planar hand poses. We also present a painting game application that enables capturing of any such dataset using a commodity depth sensor such as a Microsoft Kinect v1[1].

3. We propose a multi-layered regression method for learning from a dataset containing multiple variations. We present a method for dividing the dataset to train multiple expert regressors, each one learning from a small subset of variations. We propose a method for using regression to learn marginalization weights for multiple expert regressors [30].

4. We propose a staged probabilistic regression method, that learns expert regressors from a dataset without the explicit definition of subsets [31]. This framework learns expert regressors in stages, where evaluations of existing expert regressors are used to identify harder samples. We extend this framework to simultaneously infer hand orientation and pose. Our work is the first to show that simultaneously learning both hand orientation and pose, using 2D monocular images, enables increased accuracy for hand pose estimation as the orientation information facilitates the model to learn the hand pose representation under varying viewpoints.

1.5 Publications

The following is a list of peer-reviewed or submitted research papers developed during the duration of this doctoral research. Content from the papers in bold is included in this dissertation.

1. **Asad M., Slabaugh G., *"SPORE: Staged Probabilistic Regression for Hand Orientation and Pose Inference"*. In: Computer Vision and Image Understanding (CVIU), 2017.**

[1] https://developer.microsoft.com/en-us/windows/kinect

2. Al-Arif S.M.M.R., Asad M., Gundry M., Knapp K., Slabaugh G., *"Patch-based Corner Detection for Cervical Vertebrae in X-ray Images"*. In: Signal Processing: Image Communication (Elsevier Journal), 2017.

3. Lazareva A., Asad M., Slabaugh G., *"Learning to Deblur Adaptive Optics Retinal Images"*. In: International Conference on Image Analysis and Recognition (ICIAR), 2017.

4. Olliverre N., Asad M., Yang G., Howe F., Slabaugh G., *"Pairwise Mixture Model for Unmixing Partial Volume Effect in Multi-voxel MR Spectroscopy of Brain Tumour Patients"*. In: SPIE Medical Imaging, 2017.

5. **Asad M., Slabaugh G., *"Learning Marginalization through Regression for Hand Orientation Inference"*. In: Computer Vision and Pattern Recognition (CVPR) Second Workshop on Observing and Understanding Hands in Action, 2016. (Best Paper Award)**

6. Asad M., Yang G., Slabaugh G., *"Supervised Partial Volume Effect Unmixing for Brain Tumor Characterization using Multi-voxel MR Spectroscopic Imaging"*. In: International Symposium on Biomedical Imaging, 2016.

7. Al-Arif S.M.M.R., Asad M., Knapp K., Gundry M., Slabaugh G., *"Cervical Vertebral Corner Detection using Haar-like Features and Modified Hough Forest"*. In: International Conference on Image Processing Theory, 2015.

8. Al-Arif S.M.M.R., Asad M., Knapp K., Gundry M., Appelboam A., Reuben A., Slabaugh G., *"Hough Forest-based Corner Detection for Cervical Spine Radiographs"*. In: Medical Image Understanding and Analysis, 2015.

9. Asad M., Gentet E., Basaru R., Slabaugh G., *"Generating a 3D Hand Model from Frontal Color and Range Scans"*. In: International Conference on Image Processing, 2015.

10. **Asad M., Slabaugh G., *"Hand Orientation Regression using Random Forest for Augmented Reality"*. In: International Conference on Augmented and Virtual Reality, 2014.**

1.6 Dissertation Overview

This dissertation is organized as follows.

Chapter 1 motivates the dissertation, formulates the problem of hand orientation inference and highlights the contributions of this dissertation.

Chapter 2 presents a review of the existing state-of-the-art on hand pose estimation that can benefit from a hand orientation inference method. It also presents the existing hand orientation estimation methods and methods that marginalize multi-layered Random Forest. It details the limitations of the existing methods, putting the contributions of this dissertation into perspective.

Chapter 3 presents the preliminary concepts required for understanding this dissertation. It presents a unified framework for hand orientation regression, which forms the basis for the work presented in the subsequent chapters. It also details the Random Forest algorithm used for addressing classification and regression problems in this dissertation. We also include an overview of the data capture process along with the method used for annotating Ground Truth (GT) hand orientations. This chapter also presents the error measures used for evaluating the proposed methods.

Chapter 4 presents the hand orientation regression framework that utilizes contour-based features. We detail the framework, which utilizes two independently trained Random Forest regressors for inferring hand orientation. We also present an application of the proposed method for manipulating virtual objects in Augmented Reality.

Chapter 5 details the proposed multi-layered Random Forest method that uses marginalization through regression to learn from different variations within the hand orientation dataset. We also present a painting-based game method for reliably capturing a hand orientation dataset from multiple users. We evaluate and compare this technique with the one

proposed in Chapter 4. We also detail the application of the estimated hand orientations to enable manipulation interactions with characters in Augmented Reality.

Chapter 6 proposes a staged probabilistic regressor, that learns multiple expert regressors without explicit definition of subsets. This method uses the intermediate model evaluations to identify harder samples that are not fully learned from in the existing expert regressors. These samples are used to train the expert regressors in the subsequent stage. We extend this method to simultaneously learn both hand orientation and pose. We also present comparisons with methods proposed in Chapters 4 and 5. This chapter also describes two interaction applications enabled by the proposed orientation and pose inference method.

Chapter 7 concludes this dissertation with a summary of the proposed methods. We also explore the limitations of our work. The possible future directions that address these limitations are also discussed in this chapter.

Chapter 2

Literature Review

This chapter presents a review of the previous literature involving hand orientation and pose estimation. A significant part of our work focuses on estimating hand orientation from 2D monocular images, however, we also review existing hand pose estimation methods. These methods can be related to the single-shot hand orientation estimation, where some existing work also exploits the quantized orientation of the hand [18, 32]. A few hand pose estimation methods also indirectly estimate the orientation of the hand [19, 22, 24, 26, 27, 33, 34], however exact orientation is estimated and used by only a few [20, 25, 28, 30, 32, 35].

Existing work by Erol et al. [2] reviewed hand pose estimation methods in 2007. Following this, hand pose estimation methods have rapidly evolved as a result of the introduction of commodity depth cameras and a variety of learning-based discriminative methods [15]. A recent literature survey in [3], therefore, explores the challenges in depth-based hand pose estimation methods. Further, [3] also explores the limitations of existing hand pose datasets. This chapter builds on these existing works, where we have updated and organized the review based on the overall approach taken to address the pose estimation problem. To achieve their goals, researchers have employed the use of different modes of input data, which includes colored gloves, color and depth images [2, 3]. Based on the overall approach taken to address the pose estimation problem, we categorize previous methods into five categories, namely: generative, discriminative, hybrid, multi-layered and orientation estimation. We review these in the subsequent sections below. Following this we summarize the state-of-the-art in Table

2.1 and detail the limitations of the existing methods while motivating the work proposed in this dissertation.

2.1 Generative Methods

Generative methods use a model-based approach to address the problem of hand pose estimation. These approaches reason how different movements of a real hand result in the corresponding variations in the input data. By optimizing the parameters of a hand model to the input hand image, generative methods can simultaneously estimate the articulated hand pose and orientation. Moreover, the use of a 3D articulated hand model enables these methods to address the ambiguous cases of hand postures in 2D images [23, 26, 27].

Earlier model-based methods utilized color images to estimate hand pose, which raised a number of challenges due to ambiguity in the projected color images [23, 26, 27, 36, 37]. Furthermore, since these methods utilized a single hand model, they, therefore, lacked generalization across different users. Initial attempts at model-based hand pose estimation used simplified hand models with restricted Dof [36, 37]. These methods utilized inverse kinematics for optimizing the model, where the restricted setting resulted in limited accuracy. de La Gorce et al. [27] used generative models for both the hand and the background pixels, in a 2D image; to jointly segment and estimate the hand pose. A similar method in [26] optimized texture, illumination and articulations of a 3D hand model to estimate hand pose. This method also automatically scaled bones to adjust the hand model for tracking a new user. Wu et al. [23] used prior knowledge of hand articulations from a data glove to construct an importance sampling function. This function was then used with a sequential Monte Carlo to alternatively optimize the global and local hand motions, where the global motion was comprised of the hand orientation and translation.

Multiple cameras were employed to further increase the accuracy of generative methods [19, 39, 41, 45]. Oikonomidis et al. [45] employed a multi-camera system with eight cameras and a simplified hand model composed of transformed geometric primitives. This method utilized Particle Swarm Optimization (PSO) to minimize an objective function for estimating

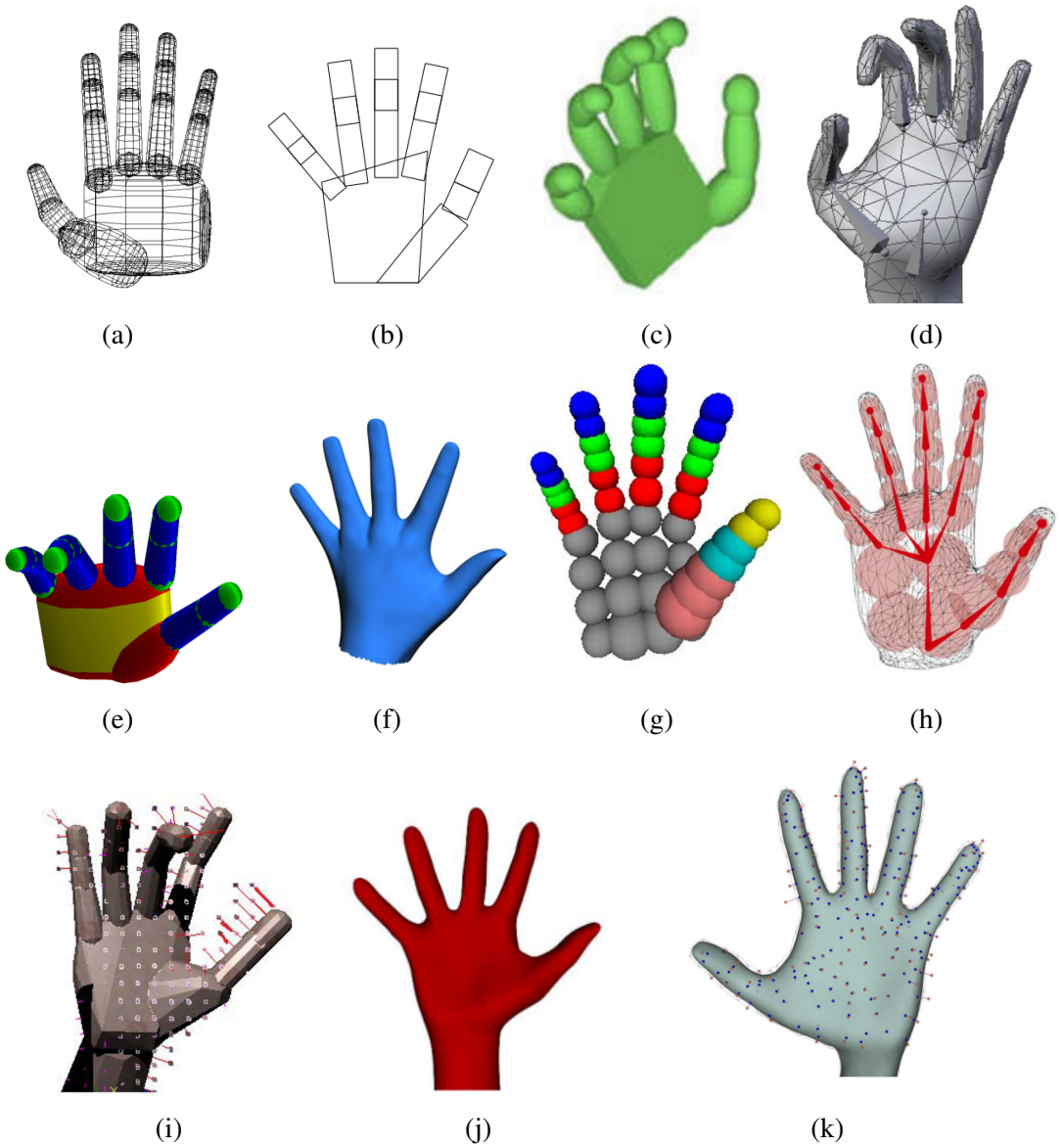


Fig. 2.1 Hand shape models, with varying complexity and level of detail, employed in generative methods. Models (a)-(d) have been used with color image-based hand pose estimation, whereas models (e)-(k) are utilized by depth-based hand pose estimation. The shown models are as follows: (a) truncated quadratics-based hand model from [36], (b) cardboard hand model from [37] and [23], (c) articulated hand model for hand silhouette synthesis in [27], (d) realistic hand model from [26], (e) simplified hand model built from transformed geometric primitives from [22], [19] and [38], (f) mesh model for modeling hand-object interaction in [39], (g) hand model approximated using 48 spheres in [40], (h) hand model with 3D Gaussians for optimization in [41], (i) simplified model used with physics constraints from [42], (j) hand shape model from [43] and [44] and (k) mesh for modeling hand shape and pose in [20].

detailed hand pose. The multi-camera-based generative method in [19] also employed eight cameras to recover hand postures in the presence of occlusion from interaction with physical objects. This method reasoned that hand-object interaction can be exploited towards a more accurate system. Ballan et al. [39] showed highly accurate tracking of two-handed hand-object interaction. This method did not focus on generalizing for multiple users, as it relied on detailed scanned meshes of hands. Furthermore, the method was computationally expensive with each frame taking 30 seconds to process. Sridhar et al. [41] used five RGB cameras and a time-of-flight (ToF) sensor to track a hand at 10 frames per second. The methods in [45], [19] and [41] utilized a person-specific hand model, limiting their applicability for a new user.

The recent introduction of depth cameras has provided researchers with additional visible 3D shape information. Furthermore, depth cameras are robust to lighting variations and background clutter. To this end, most of the recent generative methods utilize depth images along with advanced optimization techniques [3]. Oikonomidis et al. [22] applied PSO on depth images for tracking hand articulation in real-time. This method utilized a simplified hand model that was build from appropriately transformed instances of two geometric primitives i.e. cylinders and spheres [38]. The approach, however, relied on high-end GPUs. Qian et al. [40] extended this method by utilizing a fingertip-based re-initializer to assist PSO and to recover the method from tracking errors. Oikonomidis et al. [38] utilized quasi-random sampling to improve PSO, achieving speed-up of four times for single hand and eight times for two hands tracking. Melax et al. [42] inferred 3D hand poses from depth images using a simplified hand model that imposed physics constraints.

A major problem faced by generative methods is the lack of accurate 3D hand models that describes the inter-person hand shape and size variations. Hand models for some of these generative methods are shown in Fig. 2.1, where these models contain different complexity and level of detail. To address the generalization of hand models for users, some existing methods use scale calibration of a 3D hand model [23, 24, 26, 27]. Other, similar methods use simplified geometric primitives and manually adjust for different hands [19, 22, 38]. To this end, a number of methods have been proposed that estimate accurate hand models from

depth images [43, 44, 46–48]. Taylor et al. [46] proposed user-specific modeling from a rough template hand model. This method was limited by the use of a long sequence of hand motion requiring all Dof of the hand to be exercised. These limitations were addressed in [43] by learning a parametric shape and pose model using a data sequence captured from 50 different participant's hands. This enabled efficient optimization that simultaneously accounted for user-specific hand shape and general variations in pose. A similar approach was proposed in [44], where a golden energy and render-and-compare strategy enabled the method to robustly converge to the optimal solution. All these methods only worked on depth images. In contrast, Rhee et al. [47] utilized a single frontal color image of a hand to extract finger creases. These were used to localize joints and fit a 3D model with user-specific skinning. Asad et al. [48] utilized both color and depth data from a frontal scan of an outstretched pose of the hand. The approach extracted joint locations using the fingertip and inter-finger region detectors with a Naive Bayes probabilistic model. Rigid registration was performed using direct correspondences between these joint locations in the range scan and a synthetic hand model. This was followed by non-rigid registration with a thin-plate-spline deformation [49]. The resulting model fully captured the shape, size and pose variations. These hand model generation methods are computationally expensive, require specialized hardware (e.g. depth cameras) and utilize a cumbersome calibration sequence, limiting their use with generative methods on mobile devices.

Although the generative techniques discussed above are capable of estimating the underlying articulations corresponding to each hand posture, they are affected by the drifting problem, where the errors in the pose estimation are accumulated over time, which degrades the performance as the model drifts away from the actual hand pose. Moreover, optimizing the parameters with up to 27 Dof for 3D hand models is computationally expensive due to the vast optimization space [2], and in some cases requires implementation on a GPU to achieve close to real-time execution [19, 22, 45]. As a consequence, such methods are not suitable for resource-constrained devices such as wearable and mobile computers. Most of the existing generative methods require manual initialization and are not capable of recovering from

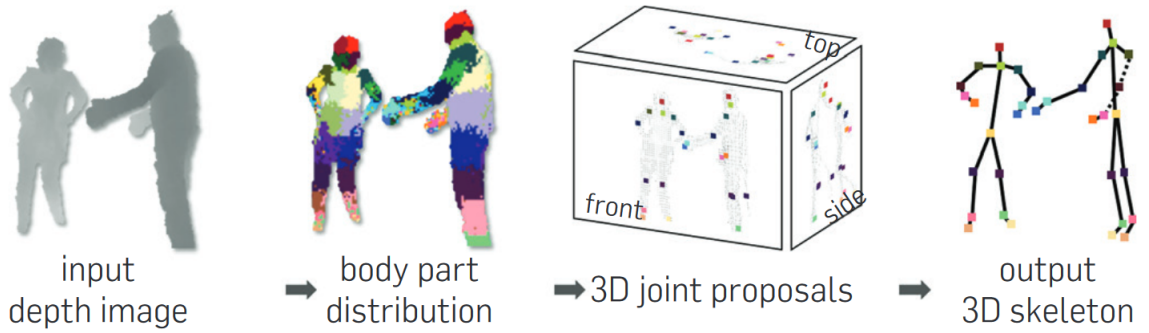


Fig. 2.2 Overview of the discriminative human body pose estimation method from [15]. Shown is an input depth image, the inferred body part distribution, 3D joint proposals and the output 3D skeleton.

tracking failures. These methods can benefit from a single-shot hand orientation and pose estimation method, that can be used to initialize as well as to correct the drifting problem.

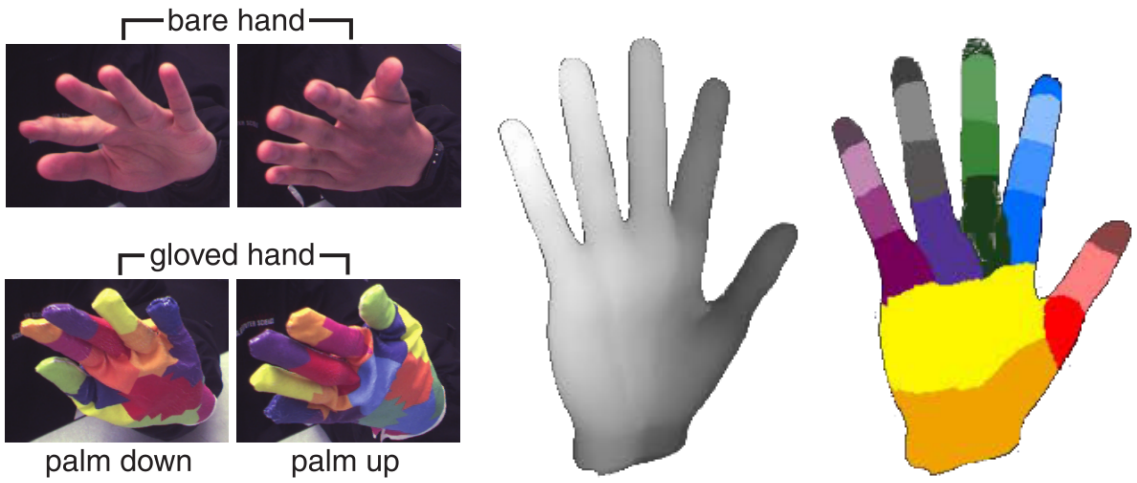
2.2 Discriminative Methods

Discriminative methods are based on a range of machine learning techniques and are able to learn the mapping from the feature space to the target label space. The ability to infer a pose from a single input image enables these methods to quickly recover from errors as compared to generative methods which tend to drift away in such scenarios [18]. This has been a major factor in their recent popularity in human pose estimation [15, 16, 50, 51]. Furthermore, these methods are computationally lightweight when compared to model-based approaches [52].

The current state-of-the-art for discriminative methods are the human body pose estimation methods [15, 16, 50]. These methods utilize depth images, where each pixel is first classified into body parts. The inferred body parts are then used to estimate the human body pose defined by 3D joint locations (as shown in Fig. 2.2). These methods employ depth comparison features which compare the depth values from nearby pixels to a threshold for classifying a pixel as a body part. This task follows a hierarchical comparison

approach, which leverages the structure of a tree in a Random Forest (RF) to combine complex threshold-based decisions. As the depth comparisons in each tree are independent from each other, unlike the different layers in convolutional neural networks (CNN), this approach can be easily parallelized for runtime efficiency [53–55]. Due to the huge success of discriminative methods in human body pose estimation, these approaches have also been utilized by a number of hand pose estimation methods [17, 18, 33, 56–58].

Wang et al. [24] presented the first discriminative method that used nearest neighbor search to infer hand pose from 2D monocular images. The approach relied on colored gloves for labeling different parts of the hand with distinct color patterns. A large synthetic dataset of hand poses, using the same pattern as the colored gloves, was generated and used in nearest neighbor search to infer the hand pose. In [56], a RF classifier was trained on a large dataset of labeled synthetic depth images to estimate the hand pose. Keskin et al. [17] improved the performance of the method in [56] by dividing the dataset into clusters and using a multi-layered Random Forest classifier. Both [56] and [17] utilized depth-based difference features originally proposed for human body pose estimation method in [15]. A number of methods extended [17] by utilizing state-of-the-art regression or classification techniques to improve accuracy for complex poses [33, 34, 57, 58]. Tang et al. [58] exploited the hierarchical relationship of different hand joints by using a divide-and-conquer strategy. This method built a topological model of the hand where the global kinematic constraints were implicitly learned. They also collected a dataset of 10 users performing various random hand postures, which they used to train and test their topological model. However, their dataset only contained limited frontal hand viewpoints as reported by [34]. Sun et al. [34] also exploited the hierarchical relationship between different parts of the hand to train a cascaded regressor. They argued that unlike the human body, the hand undergoes large variations due to changes in the viewpoint and finger articulations. They addressed this issue by presenting a 3D pixel parameterization that achieved better invariance to 3D viewpoint changes. The pixel-based classification was further improved by using spatial constraints in a Markov Random Field smoothing method [57]. This method also handled limited viewpoint



(a) (b)

Fig. 2.3 Hand labeling for discriminative methods in the literature. The methods shown are: (a) colored glove-based labeling in color images from [24] and (b) synthetic depth data generation along with the labeled hand parts from [17].

variations, however, it did not address generalization for new users due to its reliance on a synthetic dataset.

Recent interest in CNN has also been expressed in some discriminative hand pose estimation methods [59–61]. Tompson et al. [59] localized joints using CNN. They generated single-view heatmaps for joints localization using depth images as input. Ge et al. [60] extended [59] to utilize multi-view CNN. A query depth image of the hand was first projected onto three orthogonal planes to produce multi-view projections. Three CNN were then trained to infer the heatmaps of different joint locations in each projection. The inferred multi-view heatmaps were fused together to produce the final 3D hand pose. Oberweger et al. [61] explored different CNN architectures for articulated hand pose inference. They achieved this by learning the mapping of depth images onto the 3D joint locations. A regression-based joint-specific refinement stage was introduced to improve the localization accuracy.

Synthetic ground truth (GT) data generation has been widely used across discriminative methods as it is less time consuming with more accurate labels [17, 56, 57]. Fig 2.3 shows some approaches used in discriminative methods for labeling hand pose data. A major challenge faced by such methods are their lack of generalization for unseen data. Tang

et al. [18] addressed this issue by proposing a semi-supervised transductive Regression Forest for articulated hand pose estimation. This approach was able to learn hand pose from a combination of synthetic and realistic datasets of depth images. Furthermore, this method also utilized quantized viewpoints to improve pose classification accuracy. In [15], generalization for human body pose was addressed by incorporating real scenario-based variations into the synthetic data generation method. A semi-automatic method for annotating a large dataset was proposed in [62]. This method required 2D annotations on a small set of key frames. The appearance and the spatial and temporal coherence between different frames in a sequence were utilized to automate labeling of the rest of the data. The data collection process, however, was unguided, where the participants relied on descriptions of different gestures to provide variations in orientation and pose.

Recently proposed discriminative methods focus on utilizing depth images to infer detailed articulation of hands [3]. The use of depth images has been favorable as it has allowed researchers to target problems related to hand articulations, while leaving out other challenges such as segmentation and depth ambiguity in color images. A major challenge for such methods has been the collection and annotation of depth-based datasets. As noted in [3], existing depth-based datasets are diverse in terms of variation in articulations, however, many datasets contain limited variations in viewpoint. Moreover, usage of depth images has narrowed the applicability of such methods to a limited number of devices due to constraints related to depth acquisition sensors, which include form factor, cost and power consumption [21]. Technologies like Google's Project Tango [1] and Pelican Imaging [2] show the recent focus on miniaturizing the depth sensors for mobile devices. However, the need for a custom sensor with complex electronics, high-power illumination and physical constraints, such as baseline between illumination and sensor, limit the use of such devices, especially when compared to 2D monocular cameras [21]. In contrast, recent work has not seen methods that utilize color images, which are widely available across a range of devices including mobile devices. Towards this end, we propose a method that enable reliable dataset collection that utilizes the additional depth sensor information to provide ground truth orientation along

[1]https://get.google.com/tango/
[2]http://www.pelicanimaging.com/

with a corresponding color image encoding the shape, size and pose variations in hand. This enables us to study hands from a new perspective; that of orientation and pose estimation from color images without the need for camera calibration. Moreover, inspired by the exceptional performance of data-driven discriminative methods [16–18, 20, 32], we propose to apply discriminative techniques on color images. We present a more detailed analysis of the limitations of discriminative methods in Section 2.6, where we further motivate the need for the work proposed in dissertation.

2.3 Hybrid Methods

Recent literature has seen interest in utilizing a hybrid approach, that combines the generative and discriminative methods [20, 32, 59, 63, 64]. These methods utilize the one-shot pose estimation capability of discriminative models to make generative models robust to tracking failures. Moreover, the generative methods impose kinematic constraints resulting in realistically accurate descriptions of an articulated hand pose.

Xu et al. [33] took a three-step approach where they learned from a synthetic dataset of depth images. This method first estimated the in-plane orientation and 3D location of the bottom of the hand. The orientation information was then used to correct for in-plane rotation. Depth-based difference features were then used to infer a number of candidate postures of hand. Following this, a generative model was used to infer the final detailed hand pose. The resulting method turned out to be computationally expensive and was only able to generalize under in-plane rotations for a single user. Tompson et al. [59] used CNN for feature extraction and to infer heatmaps for localizing joints. Inverse kinematics was then utilized to estimate the hand pose. This approach, however, was limited by prediction of 2D joint locations, and its reliance on depth maps for determining the third coordinate, which is unavailable for occluded joints. Oberweger et al. [64] proposed a data-driven approach to estimate 3D hand poses from depth images. This method utilized CNN for estimating initial joint locations from a depth image of the hand. They replaced the generative model with a feedback loop implemented using CNN and trained to synthesize depth images from inferred

joint locations. Sharp et al. [32] utilized a discriminative re-initializer for optimizing PSO. A similar approach was proposed in [20] for hand tracking using non-linear optimization methods.

All of the emerging hybrid methods require a large dataset for learning the discriminative part, while still relying on computational resources to perform generative optimization. Owing to the complexity, such methods have not been deployed or tested on egocentric mobile devices.

2.4 Multi-layered Regression

Multi-layered regression has been previously used for hand pose estimation, where complex problems have been solved by layered regressors trained on simpler subsets of the data [17, 21, 34, 65].

Keskin et al. [17] proposed a multi-layered Random Forest (ML-RF) classification method for hand pose estimation. This method was divided into two layers, the first layer performed the shape classification, whereas the second layer used classification of individual hand parts to estimate hand pose. The final hand pose was estimated by combining the posterior probabilities for both layers as a weighted sum. The three most significant posterior probabilities from the first layer were used as weights for the corresponding posterior probabilities in the second layer. Fanello et al. [21] used a similar framework to estimate depth maps from a modified monocular 2D camera. Dantone et al. [66] proposed Conditional Random Forest for detecting facial features. Similar to [17], this method used a sum of all weighted posterior probabilities to localize facial features, given a specific head orientation. Sun et al. [67] utilized Conditional Random Forest for inferring joint locations for human body pose estimation. They argued that a multi-layered model that is conditioned on a global latent variable, such as torso orientation or human height, can significantly contribute to improved joint location prediction. A boosted classification tree, for hand shape detection with a multi-layered structure, was presented in [65]. This method performed hand detection in the first layer, whereas the second layer classified the hand into different shapes. Sun

et al. [34] utilized the hierarchical relationship between different parts of the hand to train cascaded regressors. They used a 3D pixel parameterization that achieved some invariance to 3D viewpoint variations. This showed the significance of hand orientation for articulated pose estimation. Sharp et al. [32] utilized a two-layered re-initializer, where the first layer consisted of quantized hand orientation classification and the second layer employed expert predictors to infer other elements of the hand pose. The second layer specifically contained a hand orientation refinement regressor, an offset translation regressor and a finger-based pose classifier. During prediction, the top five most likely quantized hand orientations from the first layer were used to select the corresponding distributions from expert predictors. They sampled these selected distributions and employed a PSO-based optimizer to estimate the optimal hand pose.

While the above-mentioned methods present a significant improvement in performance, their limitation comes from their reliance on the results of the first layer in the hierarchy. These systems are not capable of fully recovering from misclassifications that arise in the first layer, where their performance could significantly degrade. The weighted sum of posterior probabilities is able to partly recover errors related to these misclassifications, nevertheless, it is still influenced by the other most significant posterior. Thus, resulting in a larger model bias.

2.5 Orientation Estimation

A limited number of methods exist in the literature that estimate hand orientation [25, 28, 35, 68–70]. Most of these methods use camera calibration and hand features to build a relationship between camera pose and hand orientation. These methods do not address the generalization problem and hence require a calibration step for every new user.

Lee et al. [25] proposed a markerless camera tracking method that enabled inspection of Augmented Reality objects using fingertip tracking. This method required an offline calibration step where a sequence of images with a checkerboard marker and a coplanar outstretched hand pose were used to build a relationship between camera pose and fingertips.

This information was used for online camera pose estimation where fingertip positions were tracked using a Kalman Filter [71]. A similar method was also proposed for smart phones in [70]. Both these methods, however, relied on the assumption that the human hand remains rigid throughout the variations in orientation. Moreover, a tedious calibration step was employed for each new camera and user setting which limited widespread usage of these methods as a markerless Augmented Reality method. Mizuchi et al. [35] improved [25] by utilizing an offline calibration sequence using ARToolkit [72] markers and features based on the convexity and concavity of the hand contour. This method enhanced hand contour extraction by differentiating between adjoining fingers using elimination of low brightness regions in the segmentation mask. The convexity and concavity of the hand contour were used to extract inter-finger points and the middle fingertip. These feature points were tracked using a Kalman Filter and used along with the calibrated parameters to estimate the camera pose relative to the hand orientation. This method also assumed the use of a rigid hand pose, however, it was able to achieve some invariance to non-rigid movements using interfinger points. Nevertheless, the performance of this method was not evaluated using different variations in hands and cameras.

Ng et al. [68] improved the performance of hand-based markerless Augmented Reality methods by utilizing a stereo camera system. This method only required offline calibration of the camera setup using a checkerboard pattern and was independent of the user's hand. The method, however, relied on convexity defects of hand contour to extract fingertips and inter-finger regions. A distance transform was utilized for finding the palm center. Calibrated stereo cameras were used to find depth of each extracted feature points. The thumb fingertip, palm center and the middle fingertip were used to form a hand coordinate system. Based on the assumption that the hand remains rigid throughout orientation variations, the transformation from hand coordinate system to camera pose was found. Some robustness to feature extraction was achieved in [69], where the most dominant lines in a segmented hand region were extracted using Hough transform [73]. A homography was used along with the extracted lines to estimate camera pose.

All the existing methods focused on building a relationship between hand-based features and the calibrated camera sequence. These methods were limited by the assumption of a rigid hand pose throughout the interaction, and were unable to generalize well for a number of different hands and cameras without the need of tedious calibration steps. Moreover, these methods did not explore discriminative techniques that have been extremely successful for solving pose related problems in recent years [15, 20, 32].

To the best of our knowledge, image-based hand orientation regression has only been applied in our proposed work in [28, 30, 31]. Our method in [28] utilized two single-variate RF regressors based on an assumption that the orientation angles vary independently. This method, evaluated on a subset of hand orientation angles, showed the significance of inferring hand orientation from 2D uncalibrated monocular images. We extend the hand orientation inference framework further, in [30], by utilizing an ML-RF regression method that uses multi-variate regressors to regress the orientation angles together. We further improve the method proposed in [30] to present a staged probabilistic regressor [31]. Our proposed work does not require camera calibration which renders it suitable for a wider array of applications across different devices. The datasets used for training the proposed methods come from multiple people, which enables them to naturally handle person-to-person hand variations. Independent work proposed in [32] utilized global hand orientations from depth images to assist in hand pose optimization. The method consisted of an ML-RF, where the first layer inferred a quantized hand orientation and the second layer estimated refined orientation. The prediction probabilities, however, were utilized to sample candidate solutions for use with PSO-based optimization. The depth images provided detailed visible shape information, which introduced less ambiguities in the data, thus resulting in a less challenging orientation estimation problem in [32]. The method proposed in [74] utilized a modified Kalman Filter for tracking hand orientations from an Inertial Measurement Unit (IMU). This method, however, did not learn and generalize the mapping of image-based hand shape features onto the measured orientation of the hand.

The hand orientation inference methods proposed in this dissertation can be related to the head pose estimation method found in literature [75]. A number of different regression

methods have been previously used to estimate head pose [75–79], that learn the mapping of a feature set onto head orientation angles. These methods, however, differ from the hand orientation inference problem due to a number of reasons which include (i) fixed and distinct landmark configuration for varying head pose as opposed to rapidly changing shape and high similarity across local features (e.g. features extracted from fingers) that cannot be used as landmarks, (ii) limited self occlusion for facial features versus self occlusion that significantly affects hand shape in different viewpoints and (iii) similar configuration of facial landmarks across a number of users for head pose estimation as opposed to highly varying hand shapes and styles of doing the same hand pose across a number of users. The method described in [80] employed a RF with intensity-based difference features on grayscale images to infer head orientation. Fanelli et al. [76] also utilized RF regression for estimating head orientation from depth images. They generated synthetic annotated data using a statistical model of the human face. The RF algorithm employed depth difference features comparing two rectangular patches. This method was able to generalize well against partial occlusions and varying facial expressions.

2.6 Limitations of Existing Methods

This chapter presented the existing methods for hand pose estimation as well as hand orientation inference. The discussed methods either used a generative, a discriminative or a hybrid approach to estimate the hand pose. Moreover, these methods exploited different types of input data including color images, depth images and multi-camera systems providing a number of color images. We summarize the state-of-the-art in Table 2.1, which we assess against a number of criteria, that are directly related to the motivation of our research direction and goals. These criteria are discussed below in detail with relation to our research.

The different properties used for comparison in Table 2.1 are (i) the ability of a method to generalize for new users (GEN), whether a given approach requires (ii) model calibration (MCL) and (iii) camera calibration (CCL), (iv) direct or indirect use of hand orientation (OR), (v) guided dataset collection (GDC), (vi) methods requiring specialized hardware

such as high-end PCs (RSH) and the use of (vii) real (REAL) and/or (viii) synthetic (SYN) data for the training of discriminative methods. The execution time per frame (ET) for each method is also presented along with the method summary, however, it should be noted that the execution times are taken directly from other author's work where they all used different hardware. In most cases, the performance is measured on high-end PCs, this is highlighted in the RSH column.

The ability to generalize is vital for a given method's application in real scenarios, where a number of variations in hand shape, size and style are found along with the varying hand orientations. Overall this has been given limited importance in the existing work, where only a few methods were able to show their capabilities for new users. de La Gorce et al. [26] used 54 scaling parameters to morph a hand model during initialization. This method, however, used a number of assumptions for initialization. The discriminative method in [24] used colored gloves and scale calibration to prove its generalization capability. This method was, however, only validated on five users with no qualitative results presented. Furthermore, the author in [24] agreed that calibrating the precise shape and size of each hand would improve the accuracy of their method. In contrast to this, existing research on human body pose estimation has demonstrated solutions to generalization by inducing variations in the camera pose, body pose, shape and size, directly into the synthetic data [15]. These variations were implicitly learned by a RF classifier. A number of hand pose estimation methods followed the success of [15] to utilize synthetic depth data [17, 18, 33, 56]. Unlike the human body, hand shape varies rapidly across a number of users. Moreover, style variations and large viewpoint variations make hand pose a much more challenging task [3, 32]. Therefore, discriminative methods that utilize synthetic data were not able to report success on real hand pose datasets. Recent methods have included real datasets to show better generalization [18, 34, 58]. However, little attention has been given to generalization against hand orientation.

While model and camera calibration are important steps for most of the existing generative methods, they prove to be a major hurdle in their application to real scenarios. The dynamic nature of mobile devices induce variations in both the hand and the surrounding environment,

requiring a new calibration for each new scenario. Furthermore, achieving widespread usage of hand pose estimation methods requires them to work out-of-the-box. On the contrary, discriminative methods do not require such calibration steps and previous research has shown that they can work out-of-the-box for specific pose estimation tasks [15, 17, 18]. Furthermore, comparing the execution times for discriminative and generative methods, discriminative methods utilize minimal computational resources for real-time execution whereas generative approaches require a GPU to reach near real-time execution [22].

Fingers move with more complex articulations as compared to the head and limbs in a human body [18]. This results in self-occlusion, where different viewpoints can induce large variations for the same pose. Generative methods address this problem by optimizing against a global orientation of the hand. This, however, depends on the initialization requirement where a predefined hand orientation and pose is required. In contrast to this, discriminative methods depend on data to encode invariance to different orientations of the same pose [15, 18, 24]. Most existing discriminative methods for hand pose estimation focus on limited viewpoints [17, 56, 58]. The viewpoint is an important aspect of hand pose estimation, however, the existing datasets do not provide viewpoint variations [3, 34]. The discriminative approach proposed by Tang et al. [18] introduced a quality function that was able to combine different information gains related to the quantized viewpoints, joint articulations and hand poses. While the viewpoints, in this case, were limited, it showed the significance of having a hand pose estimation method that is invariant to changes in viewpoints. Similar performance improvement was seen in recent work that catered for hand orientation variations in hand pose estimation [18, 33, 34]. Multi-layered Random Forest-based orientation regression was used in a recent method [32]. However, this method utilized synthetic depth images which produced less ambiguous cases in the data. Hence, it addressed a less challenging orientation estimation problem as compared to estimating hand orientation from color images.

The success of discriminative methods lies in a carefully designed dataset capture process [16]. We note that such formulation is absent from most existing discriminative methods [3, 34]. Most datasets are captured with the assumption that the instructions for specific hand poses suffice for capturing it from different users. Moreover, in cases where hand orientation

variations are to be captured [18, 33, 34], participants are instructed to randomly move hands in different orientations. As we will show in Chapter 4, such an approach leads to a number of unwanted variations in style, orientations and pose. To this end, a guided method for collecting a hand orientation and pose dataset is much needed for reliable and detailed dataset collection.

Ref.	Method		Input Data			Properties						Training		Estimation Method	ET	Comments
	GM	DM	CI	DI	MC	GEN	MCL	CCL	OR	GDC	RSH	Real	Syn			
[27]	✓	✗	✓	✗	✗	✗	✓	✗	✓	✗	✓	✗	✗	Gradient Descent with Particle Filter	3min	
[26]	✓	✗	✓	✗	✗	✓	✓	✗	✓	✗	✓	✗	✗	Quasi-Newton (pose, texture and illumination)	40s	Calibration required for generalization
[23]	✓	✗	✓	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗	Importance sampling-based sequential Monte Carlo	66.67ms	Divide & conquer for orientation & pose
[22]	✓	✗	✗	✓	✗	✗	✗	✓	✓	✗	✓	✗	✗	Particle Swarm Optimization	66.7ms	Implemented on a GPU
[19]	✓	✗	✗	✗	✓	✗	✗	✓	✓	✗	✓	✗	✗	Particle Swarm Optimization	500ms	Estimates hand and interacting object pose simultaneously
[38]	✓	✗	✗	✓	✗	✗	✗	✗	✓	✗	✓	✗	✓	Particle Swarm Optimization with Evolutionary Quasi-random Search	-	Tracks multiple hands with high efficiency
[42]	✓	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	Simplified hand model-based tracking with imposed physics constraints	16.67ms	
[59]	✓	✓	✗	✓	✗	✗	✓	✗	✓	✗	✓	✓	✗	CNN for inferring heatmaps localizing joints & inverse kinematics to optimize model	24.90ms	Hybrid approach
[32]	✓	✓	✗	✓	✗	✓	✗	✗	✓	✗	✓	✗	✓	Multi-layered Random Forest & Particle Swarm Optimization	33.33ms	Hybrid approach
[20]	✓	✓	✗	✓	✗	✓	✗	✗	✓	✗	✗	✗	✓	Discriminative re-initializer & non-linear optimization using Levenberg method	-	Hybrid approach
[64]	✓	✓	✗	✓	✗	✗	✗	✗	✓	✗	✓	✓	✗	CNN for discriminative pose, synthesizing depth & updating synthesizer	2.50ms	Hybrid approach
[41]	✓	✓	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓	✓	Gradient Ascent Optimization with SVM classifier	100ms	Hybrid approach
[60]	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✓	✓	✗	Multi-view CNN learned on orthogonal plane projections of depth data	14.10ms	
[24]	✗	✓	✓	✗	✗	✓	✓	✓	✓	✗	✓	✗	✓	Nearest Neighbour	100ms	Generalization requires scale calibration of 3D kinematic model
[56]	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗	✓	Random Forest	33.33ms	
[17]	✗	✓	✗	✓	✗	✓	✗	✗	✗	✗	✓	✗	✓	Multi-layered Random Forest	-	Clustering pose specific subsets for performance gain
[18]	✗	✓	✗	✓	✗	✗	✗	✗	✓	✗	✓	✓	✓	Semi-supervised Transductive Regression Forest	40ms	Real and synthetic data fusion
[58]	✗	✓	✗	✓	✗	✓	✗	✗	✗	✗	✓	✓	✗	Latent Regression Forest with topological learning	16ms	Does not cater for viewpoint variations
[34]	✗	✓	✗	✓	✗	✓	✗	✗	✓	✗	✓	✓	✗	Random Forest with cascaded learning	3.33ms	Significance of training against viewpoint variations shown
[33]	✓	✓	✗	✓	✗	✗	✗	✗	✓	✗	✓	✗	✓	Hough Forest with generative hand model fitting	83.33ms	Only caters for in-plane rotation
[25]	✗	✗	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	Calibration using fingertips	33.33ms	Requires a calibration step

Table 2.1 State of the art for hand pose estimation. Key: GM - Generative Methods, DM - Discriminative Methods, CI - Color Image, DI - Depth Image, MC - Color images from multiple cameras, GEN - Generalization to new user, MCL - Uses Model Calibration, CCL - Requires Camera Calibration, OR - Extracts/Uses Hand Orientation, GDC - Guided Dataset Collection, RSH - Requires Specialized Hardware, Syn - Synthetic, ET - Execution time per frame.

Chapter 3

Preliminaries

This chapter presents an overview of the foundational concepts required for understanding this dissertation. A major part of our work addresses the inference of hand orientation from 2D monocular images. Therefore, this chapter starts by presenting an overview of supervised learning. As the models proposed herein utilize a probabilistic approach, we present background on different types of probabilistic outputs from a model. We then present a unified framework for learning the mapping of hand shape features from 2D hand silhouette images onto the corresponding 3D orientation. The framework puts together the key components that will be further detailed and discussed in the subsequent chapters. The common concepts, such as feature extraction and Ground Truth (GT) data annotation, are further described in this chapter. We also discuss the Random Forest (RF) algorithm, which is used for both regression and classification tasks proposed in this dissertation. We include an overview of the different dataset collection methods used to capture hand orientation and pose datasets. Furthermore, an overview of the datasets collected, for evaluating the methods proposed in this dissertation, is also provided. In the conclusion, we introduce and explain the rationale for using different error measures for evaluating the proposed work.

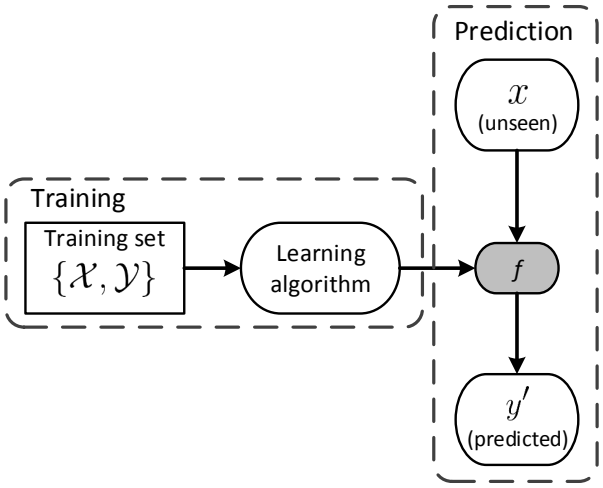


Fig. 3.1 A learning task consists of learning a mapping function f from a training set $\{\mathcal{X}, \mathcal{Y}\}$. f is used during prediction to predict target y' for an unseen input x.

3.1 Learning from a Dataset

Learning a mapping between input and labeled output variables, such that it can be used to predict the output for a previously unseen input is known as supervised learning [81]. This inference is based on the assumption that the output variable does not take a random value and that there exists a relation between input and output that can be exploited. We consider the example of hand orientation and pose estimation problem where input features describe the shape of the hand, whereas the target could either be orientation or pose. Depending on the type of target variable, supervised learning can be categorized into different machine learning tasks, namely:

Regression: Predicting the hand orientation as a function of hand shape features. The target, in this case, is continuous.

Classification: Identifying, from the shape features if a given hand is in a specific pose. The target, in this case, is categorical and can only take a single discrete value from a known set of target variables.

To understand it in detail, consider a training dataset $\{\mathcal{X}, \mathcal{Y}\}$ of input features x and target variable y [82]. Based on a learning task defined above, the target variable y can be

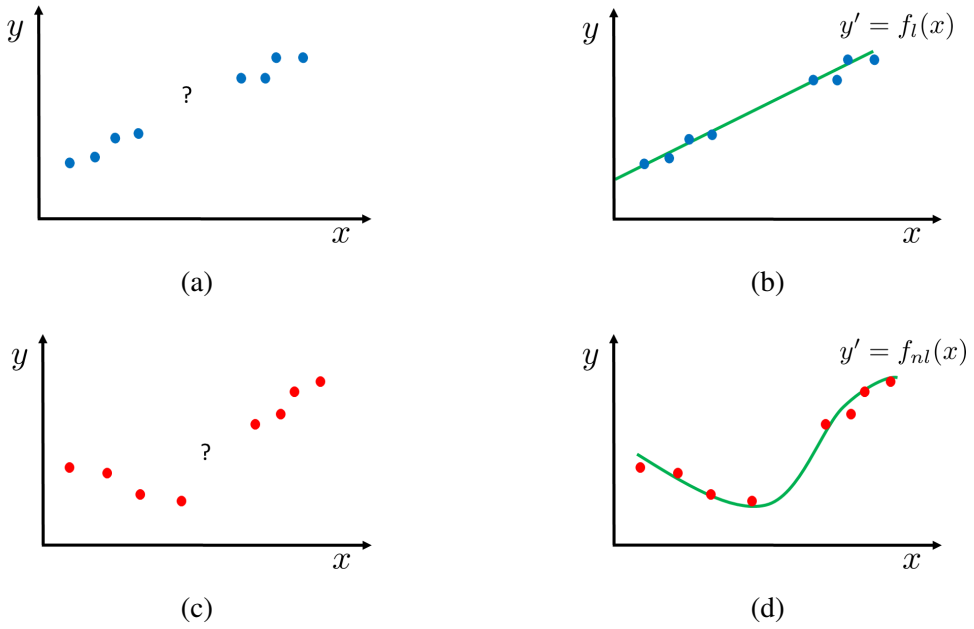


Fig. 3.2 Linear regression and non-linear regression problems, where the goal is to infer y', given an unseen input x and learned models f_l and f_{nl}. A linear regression problem is shown in (a) where (b) a linear regression model f_l can be used to learn the mapping of $x \mapsto y$. (c) shows a non-linear regression problem where a linear regression model is insufficient, therefore (d) a non-linear regression model f_{nl} can be used.

continuous or categorical. The mapping of $\mathcal{X}$ onto $\mathcal{Y}$ is learned as a function $f : \mathcal{X} \mapsto \mathcal{Y}$. Given a previously unseen input x, the function f can be used to predict a target y'. Fig. 3.1 gives a general overview of learning and prediction tasks described above.

The research in machine learning has produced an extensive literature of different learning algorithms that fit the problem posed in Fig. 3.1. Some notable examples include: (i) Linear Regression, (ii) Logistic Regression, (iii) Neural Networks, (iv) Support Vector Machines and (v) Random Forest [83–86]. The underlying formulation of any learning algorithm can be broadly categorized into two categories, namely, linear and non-linear learning [87]. A linear machine learning model learns the mapping function f as a linear combination of input features x, whereas a non-linear method builds a more complex mapping catering for non-linearities in the data. We show the concept of linear and non-linear regression with a set of abstract examples in Fig. 3.2. Fig. 3.2 (a) shows a linear regression problem, where a linear regression model f_l, shown in Fig. 3.2 (b), is able to fully learn from the data. However,

if the task is that of non-linear learning, as in Fig. 3.2 (c), then a non-linear model f_{nl} is required (Fig. 3.2 (d)).

The problem addressed in this work - that of learning the mapping $f : \{\mathbf{d}_k\}_{k=1}^{K} \mapsto \{\mathbf{o}_k\}_{k=1}^{K}$, is a non-linear regression problem where the complex relationship between hand shape features $\{\mathbf{d}_k\}_{k=1}^{K}$ and hand orientation angles $\{\mathbf{o}_k\}_{k=1}^{K}$ is learned.

3.2 Probability Distributions

A probability distribution defines how likely a random variable or a set of random variables are to be in one of their possible states. The models proposed herein take a probabilistic approach. Thus, we present an overview of the probabilistic formulation used in this work. We also describe relevant probability rules.

3.2.1 Posterior Probability

Given an input feature vector $\mathbf{x}$, the online prediction from a probabilistic model gives a posterior probability $p(y|\mathbf{x})$ of target variable y. Depending on whether the target variable y is categorical or continuous, $p(y|\mathbf{x})$ can be defined using a probability mass function (PMF) or probability density function (PDF) respectively [88].

When $p(y|\mathbf{x})$ is PMF, then it must fulfill the following conditions:

- The domain of $p(y|\mathbf{x})$ must be all possible values of y.
- The range of $p(y|\mathbf{x})$ must be in the real interval [0, 1].
- $\sum_y p(y|\mathbf{x}) = 1$.

In case of a PDF, $p(y|\mathbf{x})$ must satisfy the following properties:

- It must map from the domain of random variable y to real numbers.
- $\forall y,\ p(y|\mathbf{x}) \geq 0$.
- $\int_y p(y|\mathbf{x})dy = 1$.

3.2.2 Marginalization of Probability Distributions

In some cases, we know a probability distribution conditioned over a set of variables, e.g. $p(y|\mathbf{x},w)$ where w is an additional parameter that can be related to a learning model. We establish how such variables are introduced to learning models in Section 5.1.2. It is sometimes desirable to extract probability over just a subset of variables (such as finding $p(y|\mathbf{x})$ from $p(y|\mathbf{x},w)$). The extracted probability is known as marginal probability. Mathematically it is defined as:

$$p(y|\mathbf{x}) = \sum_{w} p(y|\mathbf{x},w). \tag{3.1}$$

Next, we describe a unified framework for learning hand orientation which forms the foundation of training and prediction frameworks proposed in Chapters 4, 5 and 6.

3.3 Unified Framework for Hand Orientation Regression

The proposed unified framework provides a general outline for designing a hand orientation regression method. The flowchart of the framework is presented in Fig. 3.3, where different specific frameworks could be designed by modifying different blocks. The framework is further divided into training and prediction stages. Training is done offline using a dataset of hand silhouette images and GT hand orientation angles. The hand orientation prediction is done online using only the hand silhouette image. Both prediction and training frameworks use the same feature extraction method to extract hand shape features from silhouette images. These are used along with the orientation angles to train a regression model. During prediction, the hand shape features, extracted from an unseen hand silhouette image, are used in the trained model to infer the corresponding hand orientation. This dissertation makes use of some necessary assumptions to achieve the research goals, which are discussed next. Following this we describe each block within the proposed unified framework.

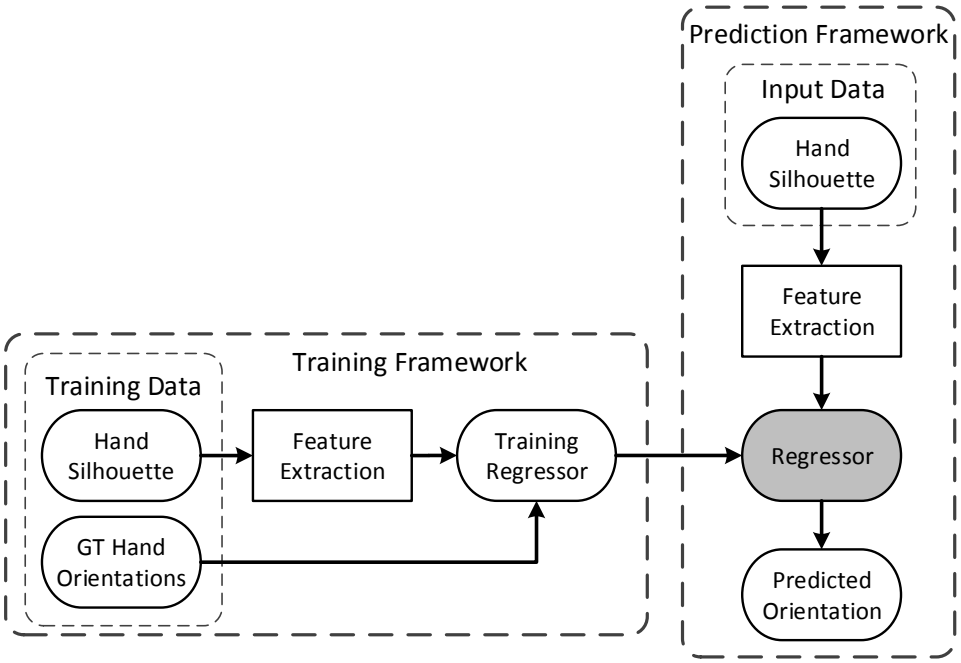


Fig. 3.3 Flowchart of the proposed unified framework, showing the training and prediction frameworks for hand orientation regression. The trained regressor is highlighted with a solid gray background.

3.3.1 Assumptions

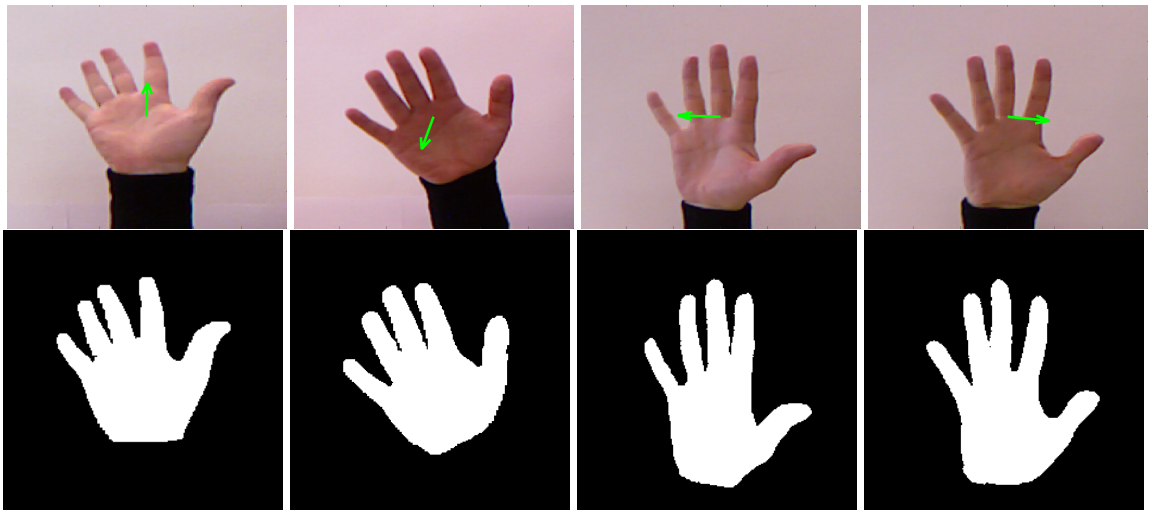
Most mobile devices are equipped with 2D monocular cameras. 3D depth cameras are not widely available on such devices due to their high power consumption, cost and relatively larger form-factor [21]. Our proposed hand orientation estimation methods are targeted for mobile devices, and for this reason, we only use 2D monocular images. Most existing state-of-the-art methods utilize depth data, where the focus is to infer detailed articulated hand pose [17, 18, 22]. These methods are not suitable for a mobile scenario where, in addition to the absence of depth sensors, limited computational resources are available. The proposed methods for hand orientation estimation assume the use of 2D monocular cameras, where limited computational resources are available and real-time performance is required.

We assume that the hand orientation can be represented with a single 3D normal vector for a planar hand pose. This enables us to reliably extract hand orientation angles encoded by the 3D normal vector, which is satisfied by a limited set of articulated hand postures. Nevertheless, such assumption facilitates our research to focus on the effects of hand orientation variations

Fig. 3.4 Hand images with orientation angles in the range $\sqrt{\phi^2 + \psi^2} \gg 45°$. The large orientations result in self-occlusion where the visible shape of the hand is significantly occluded. Such orientations are not addressed in this dissertation.

with a predefined set of planar hand shapes. This dissertation refers to planar hand shapes as hand poses, where our aim is to study the effects of orientation variations on such hand poses. While the problem seems similar to pose estimation for rigid objects, it is quite different from it as our data contains multiple sources of variations. These include inter-person hand shape and size variations and intra-person pose and style variations.

Given the 3D normal vector, we extract the orientation encoded by azimuth (ϕ) and elevation (ψ) angles. Our aim is to model variations in orientations for fronto-parallel hand, therefore we limit the orientation angles to $\sqrt{\phi^2 + \psi^2} \leq 45°$. On the contrary, hand orientations with $\sqrt{\phi^2 + \psi^2} \gg 45°$ are affected by self-occlusion where the visible shape of the hand is significantly occluded. Moreover, our ground truth orientation extraction method (described in Section 3.5.2) restricts the orientations variations to a fronto-parallel hand. Fig. 3.4 shows some example hand images where orientation angles are $\sqrt{\phi^2 + \psi^2} \gg 45°$. It can be seen that in such cases self-occlusion presents a challenge for extracting both reliable ground truth orientation angles and shape features that are necessary for training our proposed methods.



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

Fig. 3.5 Symmetry problem in silhouette images shows color images of the hand in the top row and the segmented silhouette images in the bottom row. The absence of depth information in silhouette images results in similar silhouettes for symmetrically opposite hand orientations. The images in (a)-(b) and (c)-(d) are from symmetrically opposite orientations of the hand. GT normal vector is shown as green arrow.

Skin and hand segmentation have a long history in computer vision, where many segmentation techniques have been devised [89–91]. We, therefore, assume the segmentation problem is already solved. This enables us to focus on the hand orientation estimation problem, given a segmented silhouette image of planar hand shape. While the silhouette images facilitate capturing the hand shape variations required for learning the problem of hand orientation inference, they are also affected by the symmetry problem, where the absence of depth information results in symmetrically opposite orientations to produce similar hand shape as shown in Fig. 3.5.

To robustly extract hand shape features, we assume that the in-plane orientation θ of the hand will always be within a predefined range of an upright hand pose, where $\theta = 90°$. Our assumption is satisfied by setting the operating range on the in-plane orientation to be $0° < \theta < 180°$.

3.3.2 Feature Extraction

A rich body of literature exists that provides a number of feature extraction techniques for hand pose estimation [17, 18, 59, 61, 92, 93]. Recent work has focused on utilizing depth images from commodity depth cameras, where depth difference features, similar to the one proposed in [16] for human body pose estimation, have been used [17, 18]. Some methods have also employed Convolutional Neural Networks (CNN) for representation learning on depth images, where the most suitable convolutional kernels have been learned [59, 61]. However, these CNN-based methods are limited by the requirement for a large training dataset and specialized hardware with enough computational resources [94]. We aim to learn from color images which, unlike most recent methods that utilize depth images, presents a number of challenges including lighting variations, inter-person skin color variations and scale changes across different hands and scenes. Moreover, our research studies hand orientation variations in color images using machine learning models, which has not been previously studied. We observe that the changing orientation of the hand induces changes in the projected hand shape. We, therefore, utilize contour-based features in our work as these features encode the geometric hand shape variations that directly correspond to changes in orientation of the hand [28]. Similar features have been previously used for hand shape-based gesture recognition [92] and person recognition [95]. As we will show in this dissertation, these features also prove sufficient for jointly learning hand orientation and pose. Moreover, we note that the hand contour is more robust to scene illumination and compactly encodes (as a 1D signal) the hand's global orientation unlike local feature descriptors like texture, shape context, or SIFT [93, 96]. The primary goal of this work is hand orientation estimation, whereas we also address the hand pose estimation as a secondary objective. Nevertheless, changes in the CDF relate to variations in both hand orientation and pose. Moreover, we also employ a method for aligning and normalizing the extracted features. The method for extracting CDF vectors is described in the following section.

Contour Distance Features

Given a dataset $\{\mathbf{s}_k\}_{k=1}^{K}$ of input silhouette images, we propose a method to compute a corresponding CDF set $\{\mathbf{d}_k\}_{k=1}^{K}$. The contour extracted from each silhouette image in $\{\mathbf{s}_k\}_{k=1}^{K}$ consists of points $\mathbf{p}_k = \{\mathbf{p}_{k1}, \cdots, \mathbf{p}_{ki}, \cdots \mathbf{p}_{kI}\}$, where k specifies the sample index and i is the index for each point in the contour. Let a contour distance for a single silhouette image be denoted by $\widetilde{\mathbf{d}}_k = \left\{\widetilde{d}_{k1}, \cdots, \widetilde{d}_{ki}, \cdots \widetilde{d}_{kI}\right\}$. $\widetilde{d}_{ki}$ is computed by calculating the Euclidean distance of each of the contour points $\mathbf{p}_{ki} = \{p_{ki}^x, p_{ki}^y\}$ to a prevalent point on the wrist $\mathbf{q}_k = \left\{q_k^x, q_k^y\right\}$ and is given by:

$$\widetilde{d}_{ki} = \sqrt{\left(q_k^x - p_{ki}^x\right)^2 + \left(q_k^y - p_{ki}^y\right)^2}, \tag{3.2}$$

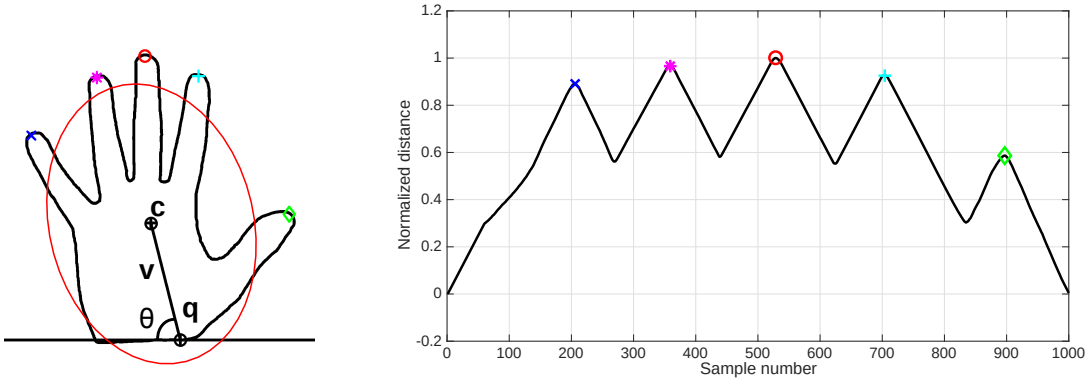
where $\mathbf{q}_k$ is a prevalent point on the wrist that is extracted, for each sample in $\{\mathbf{s}_k\}_{k=1}^{K}$, by emanating a ray from centroid in the direction of the wrist. We further discuss the approach for extracting $\mathbf{q}_k$ in the next section. The extracted features have different number of samples and magnitude depending on the scale changes and inter-person hand shape variations. We normalize the magnitude as:

$$\overline{\mathbf{d}}_k = \frac{\widetilde{\mathbf{d}}_k}{\max\limits_{1 \leq i \leq I} (\widetilde{\mathbf{d}}_{ki})}. \tag{3.3}$$

$\overline{\mathbf{d}}_k$ is then resampled to a specified number of samples Υ to produce $\mathbf{d}_k \in \{\mathbf{d}_k\}_{k=1}^{K}$. In our experimental evaluation, we found that the size of $\overline{\mathbf{d}}_k$ is related to the scale of the hand, which we found to be in the range $800 - 1400$ samples. We empirically choose $\Upsilon = 1000$ to preserve the variations in the feature vector.

Extraction of a Prevalent Point on the Wrist

We now describe the method for extracting a prevalent point $\mathbf{q}_k$ on the wrist in a silhouette image $\mathbf{s}_k$. This point is used as a reference point in Equation 3.2 to extract the CDF vector. Furthermore, the point $\mathbf{q}_k$ also aligns the corresponding CDF vector. Fig. 3.6 shows the method for extracting such prevalent point, for a given hand contour, along with its corresponding CDF vector. We use the in-plane orientation θ of the hand which can be defined by the angle between the x-axis and the major axis of an ellipse that fits the hand contour.



(a) (b)

Fig. 3.6 CDF vector extraction from hand contour showing (a) the method for extraction of prevalent point $\mathbf{q}_k$ on the wrist using a fitted ellipse with in-plane orientation θ, centroid $\mathbf{c}_k$ and a ray $\mathbf{v}_k$ and (b) the corresponding CDF vector.

Ellipse fitting is achieved using Matlab region properties function[1] which uses the method proposed in [97] to find an ellipse that has the same normalized second central moments as the hand contour. Given θ and the contour centroid $\mathbf{c}_k$, an equation of a ray emanating from $\mathbf{c}_k$ can be defined by:

$$\mathbf{v}_k = \xi \kappa \hat{\mathbf{v}}_k + \mathbf{c}_k, \tag{3.4}$$

where $\hat{\mathbf{v}}_k$ is the unit vector encoding the direction as:

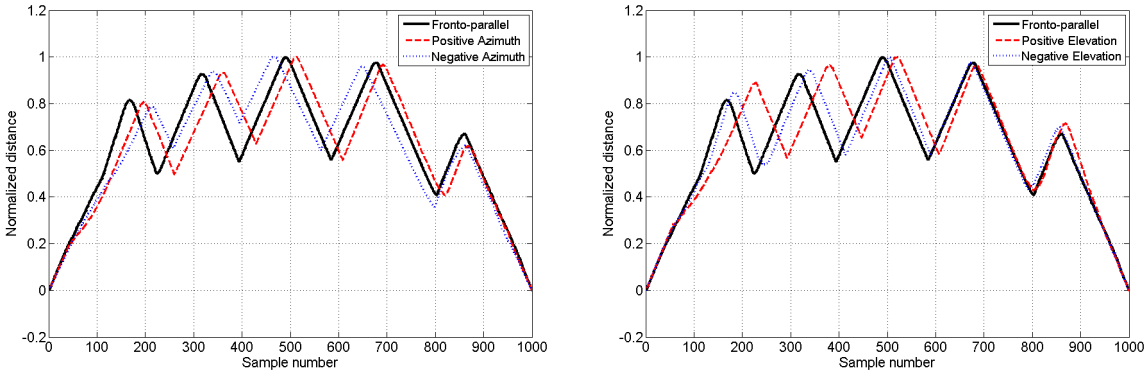
$$\hat{\mathbf{v}}_k = \frac{\begin{bmatrix} 1 \\ \tan\theta \end{bmatrix}}{\sqrt{1^2 + \tan^2\theta}}, \tag{3.5}$$

ξ is a scalar for correcting the direction of $\hat{\mathbf{v}}_k$ defined as:

$$\xi = \begin{cases} +1 & \text{if } \theta < 90°, \\ -1 & \text{if } \theta \geq 90°, \end{cases} \tag{3.6}$$

and κ is a parameter that changes the length of the ray.

[1] https://uk.mathworks.com/help/images/ref/regionprops.html



(a) (b)

Fig. 3.7 Variations in the CDF vectors with varying orientations in (a) azimuth (ϕ) axis and (b) elevation (ψ) axis only.

The direction scalar ξ is calculated using Eq. 3.6 based on the assumption that the in-plane orientation θ of the hand will always be in the range $0° < \theta < 180°$. ξ is used in Equation 3.4 to correct the direction of the ray $\mathbf{v}_k$ so that it is always propagating towards the wrist. Our proposed method increases κ until the ray intersects with the contour at a point $\mathbf{q}_k \in \mathbf{p}_{ki}$ on the wrist. This point is also used as a starting point for the distance feature calculation.

Fig. 3.7 shows the variations in CDF vectors corresponding to different hand orientations, which are exploited in our work. For better visualization, we only show the CDF vectors for orientations near the ends of our defined orientation space. The corresponding hand contours and images are shown separately in Fig. 3.8, along with the normal vector encoding the hand orientation. The direct hand shape changes for each angle combination can be observed from this figure. As further discussed in the following chapters, we are interested in improving the non-linear regression, to learn the mapping of CDF onto corresponding hand orientations, by proposing a number of probabilistic regression approaches.

3.4 Why Random Forest?

Random Forest (RF) has previously been used for fast and robust human pose estimation for the full-body as well as the hand from depth images [15, 56]. Previous studies have

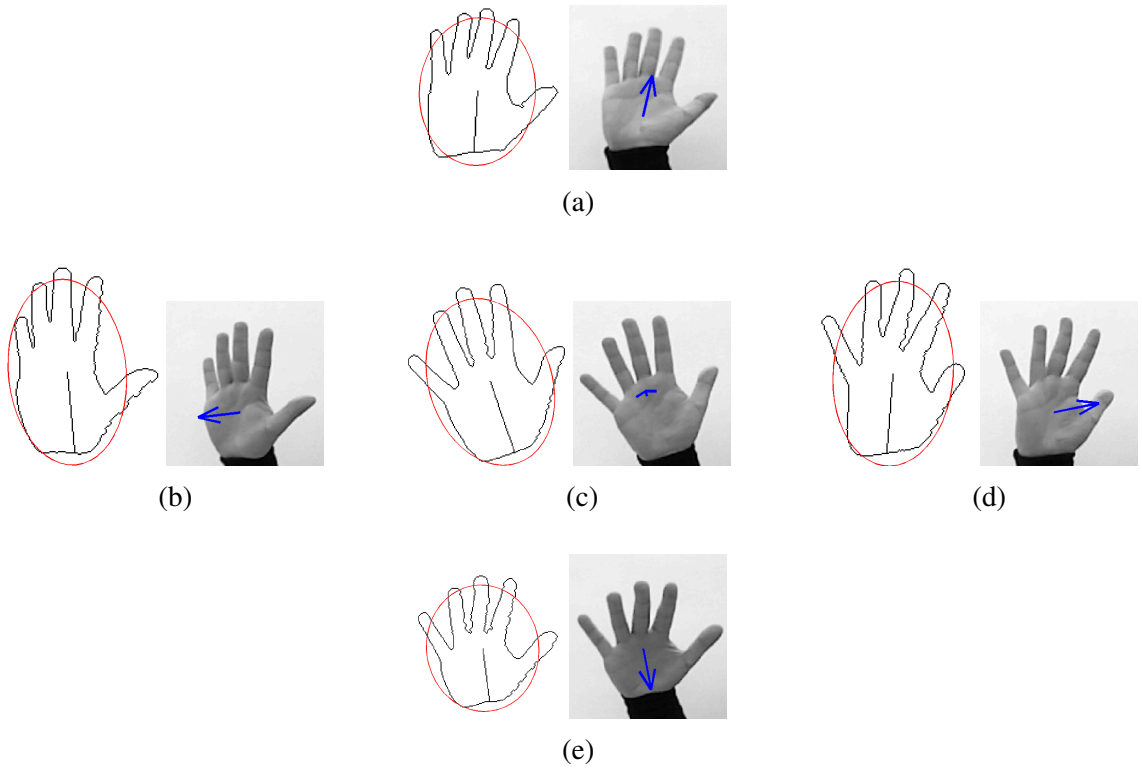


Fig. 3.8 Variations in the contour corresponding to the CDF vectors in Fig. 3.7 along with the normal vector encoding the GT azimuth (ϕ_k) and elevation (ψ_k) orientation angles. From Fig. 3.7 each plot corresponds to the contours in the following manner: (a) Negative Elevation: $\phi_k = +12.60°$ and $\psi_k = -38.96°$, (b) Negative Azimuth: $\phi_k = -36.67°$ and $\psi_k = +8.59°$, (c) Fronto-parallel: $\phi_k = +2.29°$ and $\psi_k = -0.57°$, (d) Positive Azimuth: $\phi_k = +47.56°$ and $\psi_k = +2.29°$ and (e) Positive Elevation: $\phi_k = +10.31°$ and $\psi_k = +41.83°$.

shown a number of desired properties of RF for both classification and regression [86, 98–100]. Fernandez et al. [98] evaluated performance of 179 classifiers arising from 17 base classification methods, including Neural Networks, Support Vector Machines and Random Forest, on 121 different datasets. Their study showed that Random Forest was the best classifier, outperforming other classification methods on a variety of publicly available datsets. This section discusses the rationale for choosing RF over other methods for addressing the problem of hand orientation and pose estimation.

RF is an ensemble of Randomized Decision Trees, which are non-parametric. They enable learning the non-linear and complex relationship between input and output variables. Their empirically evaluated performance has shown them to outperform most of the state-

of-the-art methods for high dimensional data problems [101]. Given its performance, RF is considered to be close to an ideal learner [94]. Each Decision Tree is independently trained, where the induced randomization ensures diversity in the Forest, resulting in better generalization for unseen input while preventing over-fitting.

The tree structure in RF is capable of inherently addressing multi-variate and multi-class problems. Compared to classifiers like Support Vector Machine (SVM) or Adaboost, RF is more efficient as it does not rely on decomposition of multi-class problems into binary classification tasks [102, 103]. Moreover, the decision boundaries generated by RF have maximum margin properties similar to SVM [99]. Our work undertakes the classification of a hand shape into pose defined by a set of previously seen postures. Therefore, the multi-class property is encouraging.

The split node optimization implicitly does feature selection, facilitating efficient learning from high dimensional data without the need for dimensionality reduction preprocessing. Our initial work presented in Chapter 4 utilizes a dimensionality reduction preprocessing. The same procedure step is not used in Chapters 5 and 6, where RF on high dimensional data produces more accurate models. Moreover, the learned features and the predictions are interpretable, enabling a better understanding of the learned mapping. This facilitates research and development of robust model configurations related to a specific problem.

The RF can learn from continuous as well as categorical target labels for solving regression and classification problems, respectively. Furthermore, with some modification, both classification and regression problems can be combined into one framework. We will take advantage of this feature of RF in Chapter 6 to simultaneously infer hand orientation and pose.

Our proposed marginalization methods in Chapters 5 and 6 follow a probabilistic formulation. In Chapter 6 we demonstrate that these methods are generalizable and can potentially work with any probabilistic regression model. Using RF, the terminal nodes in a tree can be used to make probabilistic predictions over the target variable, which significantly contributes to the development of the proposed marginalization methods.

One of the most beneficial properties of RF is that the Decision Trees are independent from each other. It helps in both training and testing, where parallel processing power can be used to achieve high computational efficiency. Furthermore, owing to the simplicity of the algorithm, a number of opensource implementations exist that are targeted for mobile devices or resource-constrained devices [2] [3].

A disadvantage of using RF is that it lacks a theoretical background, as due to the induced randomness it is not possible to mathematically formulate the underlying algorithm. This is one of the reasons why sometimes choosing different parameters empirically cannot be avoided. Another drawback of RF is the conventional feature extraction and learning framework. Deep learning methods have recently gained popularity as they enable joint learning of feature representation and classifiers [55, 104]. This architecture has outperformed previous state-of-the-art that used feature extraction as a separate step [55]. Moreover, the success of deep learning is demonstrated by the fact that these methods can even outperform humans on the task of image classification [105]. However, the deep learning methods are limited by the requirement for large training datasets and specialized hardware with enough computational resources [94].

The following section presents the employed training and prediction algorithms for the RF model.

3.4.1 Classification and Regression using Random Forest

In our work, we utilize RF for both classification and regression tasks. The classification and regression forests only differ in terms of the estimation of information gain and leaf node probability distributions. Further in-depth literature on RF can be found in [99].

The forest is a collection of T trees which are trained using a training dataset $\mathcal{U} = \{(\mathbf{d}_1,\mathbf{l}_1),\cdots,(\mathbf{d}_k,\mathbf{l}_k),\cdots,(\mathbf{d}_K,\mathbf{l}_K)\}$, where $\mathbf{d}_k$ is the CDF vector with the corresponding label $\mathbf{l}_k$. The label $\mathbf{l}_k$ can be categorical or continuous for classification or regression respectively. It can also be single-variate or multi-variate. However for sake of generality, this section

[2] https://github.com/mpielot/androidrf
[3] https://github.com/karpathy/forestjs

presents the details with a multi-variate representation. When learning the mapping of CDF vectors onto hand orientation angles, we replace the label $\mathbf{l}_k$ with the orientation angle vector $\mathbf{o}_k = (\phi_k, \psi_k)$. Each tree in the RF consists of split nodes, each responsible for performing a binary split on the input dataset, and terminal leaf nodes that store the probability distribution of the data propagated down the branches of the tree. We use random splitting criteria from [106], where splitting parameters are learned by random selection of test features and corresponding thresholds. This approach is computationally efficient with better overall accuracy. The learned parameters $\Theta = (w, \tau)$ are stored at each split node, where w is the index of the test feature and τ is its corresponding learned threshold defining the split. The data arriving at the j^{th} node is divided using a splitting function $f\left(\mathcal{U}_j, \Theta\right)$ defined as:

$$f\left(\mathcal{U}_j, \Theta\right) = \begin{cases} Left & \text{if } \mathcal{U}_j(w) < \tau, \\ Right & \text{otherwise.} \end{cases} \tag{3.7}$$

Driven by maximizing the information gain $Q\left(\mathcal{U}_j, \Theta\right)$, the splitting function divides the data into two sets $\left\{\mathcal{U}_j^{Left}, \mathcal{U}_j^{Right}\right\} \in \mathcal{U}_j$ for the child nodes:

$$Q\left(\mathcal{U}_j, \Theta\right) = H\left(\mathcal{U}_j\right) - \sum_{b \in \{Left, Right\}} \frac{\left|\mathcal{U}_j^b\right|}{\left|\mathcal{U}_j\right|} H\left(\mathcal{U}_j^b\right), \tag{3.8}$$

where $H\left(\mathcal{U}_j\right)$ is the entropy of $\mathcal{U}_j$, which is defined for a multi-class classification problem as:

$$H\left(\mathcal{U}_j\right) = -\sum_{\mathbf{d} \in \mathcal{U}_j} p(\mathbf{l}|\mathbf{d}) \log p(\mathbf{l}|\mathbf{d}). \tag{3.9}$$

For a multi-variate regression problem, $H\left(\mathcal{U}_j\right)$ is defined by differential entropy as:

$$H\left(\mathcal{U}_j\right) = -\frac{1}{|\mathcal{U}_j|} \sum_{\mathbf{d} \in \mathcal{U}_j} \int_{\mathbf{l}} p(\mathbf{l}|\mathbf{d}) \log p(\mathbf{l}|\mathbf{d}) d\mathbf{l}. \tag{3.10}$$

The conditional probability $p(\mathbf{l}|\mathbf{d})$ is defined as:

$$p(\mathbf{l}|\mathbf{d}) = \mathcal{N}\left(\bar{\mathbf{l}}, \Sigma_l\right), \tag{3.11}$$

where $\bar{\mathbf{l}}$ is the mean and Σ_l is the covariance of all target labels arriving at j^{th} split node [99, 107]. Substituting Equation 3.11 in Equation 3.10 yields the following differential entropy:

$$H\left(\mathcal{U}_j\right) = \frac{1}{|\mathcal{U}_j|} \sum_{\mathbf{d} \in \mathcal{U}_j} \log(|\Sigma_l|). \tag{3.12}$$

The branches in the tree terminate with leaf nodes that contain the probability distributions of the data arriving as a result of the above splitting process. During the online prediction, given an input feature vector $\mathbf{d}_k$, each tree gives a posterior probability $p_t\left(\mathbf{l}|\mathbf{d}_k\right) \forall\, \mathbf{l}$ which is aggregated over all the trees as:

$$p\left(\mathbf{l}|\mathbf{d}_k\right) = \frac{1}{T} \sum_{t=1}^{T} p_t\left(\mathbf{l}|\mathbf{d}_k\right), \tag{3.13}$$

where the final value of $\mathbf{l}$ is determined by maximum likelihood estimate (MLE) [87]:

$$\mathbf{l}^* = \arg\max_{\mathbf{l}} p\left(\mathbf{l}|\mathbf{d}_k\right). \tag{3.14}$$

3.5 Data Capture

The existing methods for articulated hand pose estimation do not generalize for a number of users under varying hand orientations [3]. To this end, the existing datasets are limited in terms of hand shape, size and style variations that arise as a result of different users performing the same task. Furthermore, the existing methods in literature lack a method for annotating a dataset with global hand orientation. This motivates us to propose a method for capturing and annotating reliable hand orientation dataset using a commodity depth sensor. The details of our approach are presented below.

3.5.1 Color and Depth Images

We use Microsoft Kinect sensor v1[4] to capture and annotate a hand orientation and pose dataset. We make use of Kinect v1 due to its availability in our research group, however we note that the methods proposed in this dissertation are applicable to any sensor that captures depth and color images. The Kinect sensor can be used to capture aligned 2D color and 2.5D depth images. Fig. 3.9 visualizes these two modes of images with a planar hand pose. The color image shown in Fig. 3.9 (a) stores the captured color information in three channels, i.e. RGB representing Red, Green and Blue channels. The depth image shown in Fig. 3.9 (b) encodes the visible shape information of the hand, represented as the distance of each pixel from the sensor. Fig. 3.9 (c) and (d) visualize the depth image using a colormap and a 3D point cloud, respectively. The acquired depth information enables us to reliably annotate the 3D hand orientation for a captured color image such as the one shown in Fig. 3.9 (a). We further describe the method for extracting the hand orientation angles from depth images of planar hand poses in the next section.

3.5.2 Ground Truth Orientation Annotation using Depth Images

The proposed hand orientation regression frameworks are evaluated on datasets that contain silhouette images and GT orientation angles. These GT orientation angles are only used during the training phase and are extracted from aligned depth image of hand by fitting an equation of a plane. In our research, we use planar hand poses for reliable and accurate hand orientation annotation. These planar poses are sufficient for showing the contribution of the proposed methods. Given a depth image of hand, we can extract the 3D point cloud shown in Fig. 3.9 (d), where coordinate system of depth sensor is used. We use RANdom SAmple Consensus (RANSAC) to fit an equation of a plane on this 3D point cloud [108] defined as:

$$n_0 = xn_x + yn_y + zn_z, \tag{3.15}$$

[4]https://developer.microsoft.com/en-us/windows/kinect

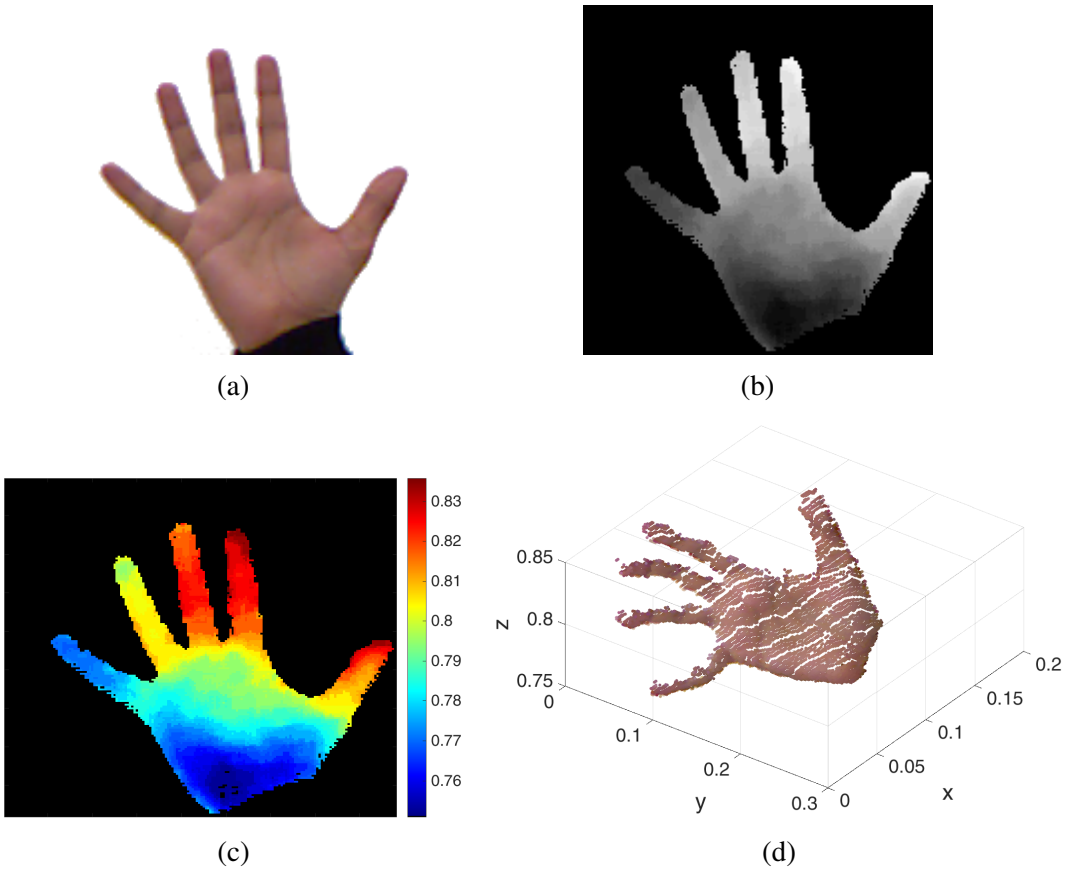


Fig. 3.9 Color and depth images captured from a Microsoft Kinect v1 sensor. The captured data consists of (a) 2D color image and (b) 2.5D depth image of the hand. The depth image shows the 3D shape information measured in distance in meters from the sensor. (c) visualizes the depth information, whereas (d) shows the 3D point cloud.

where the individual coefficients form a normal vector $\mathbf{n}$ such that:

$$\mathbf{n} = [n_x, n_y, n_z]^T, \tag{3.16}$$

which is used to calculate the corresponding GT orientation angles as:

$$\phi_k = \cos^{-1} n_x, \;\; \psi_k = \cos^{-1} n_y, \tag{3.17}$$

where ϕ_k and ψ_k are GT azimuth and elevation angles respectively. Fig. 3.10 shows how these angles are measured from their respective axes.

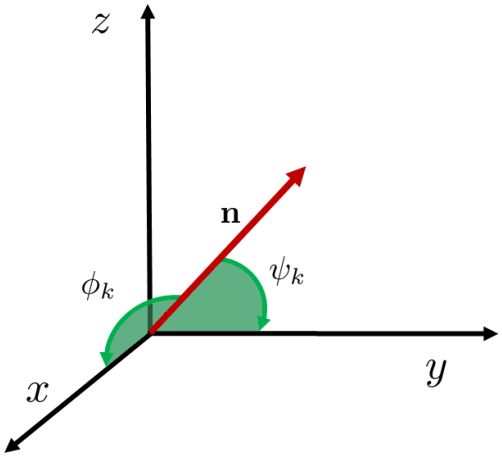


Fig. 3.10 Azimuth (ϕ_k) and elevation (ψ_k) angles as measured from x and y axes, where the coordinate system of depth sensor is used. The in-plane orientation is estimated using the orientation of ellipse fitted to the hand silhouette image.

3.6 Captured Datasets

We use the discussed dataset annotation method to collect a number of hand orientation and pose datasets. Table 3.1 details the four different datasets collected in our research. Each dataset focuses on validation against a number of different challenges faced by the learning methods, which include generalization against different shape, size and style as well as variations in hand orientation and pose. Fig. 3.11 shows variations in style, shape and size of 15 different hands from our dataset with the same hand orientation, whereas Fig. 3.12 shows the hand shape variations, for a single user, within our defined orientation space. *Dataset 1* is collected by instructing participants to vary hand orientation along both azimuth and elevation axes. This dataset is captured in an unguided way which means that during the data collection no feedback was available to users. We note that each participant has a different style of performing different orientation movements, due to which an unguided method produces inconsistency in orientations within the collected data. *Dataset 1* is used in our initial hand orientation work presented in Chapter 4. We propose a novel painting game application that leverages a real-time feedback system for guiding the participants

Dataset	*Dataset 1*	*Dataset 2*	*Dataset 3*	*Datatset 4*
Used for	SL-RF SV orientation	ML-RF MtR orientation	SPORE orientation	SPORE orientation and pose
Number of participants	13	15	22	10
Number of samples	1624	7059	9414	8675
Sampled space	$\pm 10°$ along both axes	All defined space	All defined space	All defined space
Coverage	$-45°$ to $+45°$	$-45°$ to $+45°$	$-45°$ to $+45°$	$-45°$ to $+45°$
Number of Poses	1	1	1	4
Collection method	Unguided	Guided	Guided	Guided
Chapter used	4	5	6	6

Table 3.1 Details of the datasets collected for experimental validation of the work proposed in this dissertation.

in the defined hand orientation space and to cover the complete orientation space (detailed in Section 5.1.4). It enables some level of control of participants' movements as we can guide them to provide consistency in orientation variations. This method is used to collect *Dataset 2*, *3* and *4*. *Dataset 2* is used for validation of our proposed marginalization through regression method in Chapter 5, whereas the staged probabilistic regression method proposed in Chapter 6 is validated with *Dataset 3* and *4*. In Chapter 6, we present an extension of our work to simultaneously infer hand orientation and pose. To evaluate the performance of this extension, we collect the *Dataset 4* with four poses where for each pose the painting game application is used to acquire data with consistent hand orientation variations. We describe different methods used for data collection along with their respective work in the following chapters.

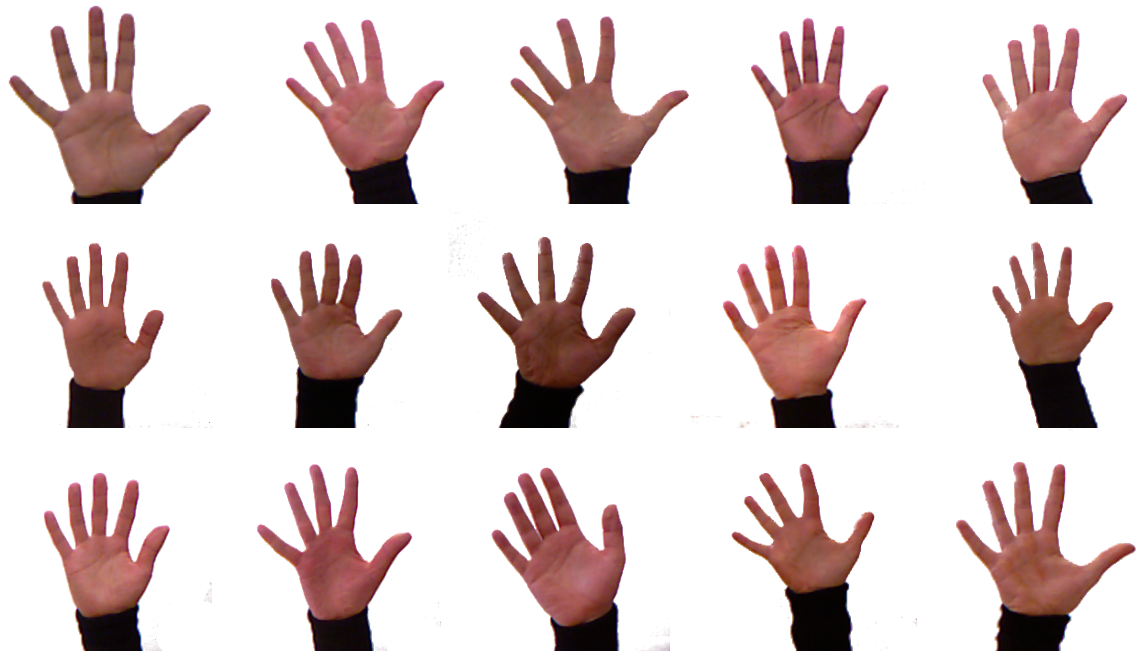


Fig. 3.11 Variations in style, shape and size of hand from 15 participants in our dataset. The hand images are shown for the same orientation.

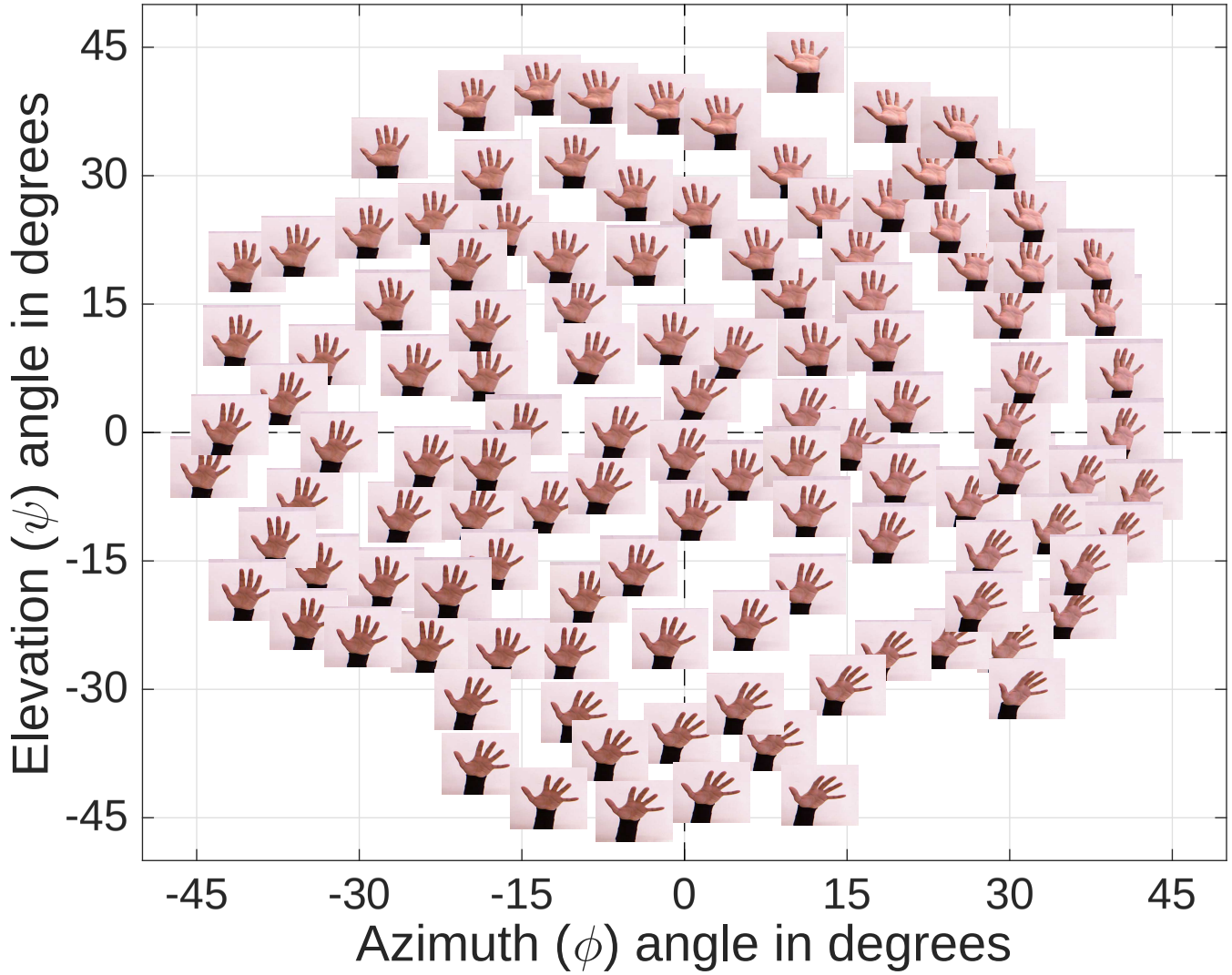


Fig. 3.12 Hand shape variations corresponding to variations in the hand orientation angles from a single user. We show the orientation angles in our defined orientation space of $\sqrt{\phi^2 + \psi^2} \leq 45°$ for a single user.

3.7 Error Measures

A major part of this dissertation presents regression methods that deal with the inference of continuous hand orientations. We evaluate the proposed methods using a number of qualitative as well as quantitative error measures. These include Mean Absolute Error (MAE) for each orientation angle, Combined Mean Absolute Error (CMAE) for both azimuth and elevation angles, GT versus predicted angle plots and percentage data versus error plots. We present a brief overview of these measures below.

3.7.1 Mean Absolute Error

Given a set of GT orientation angles (ϕ_k, ψ_k) and the corresponding predicted angles (ϕ_{pk}, ψ_{pk}) from a trained regressor, the MAE (ϕ_m, ψ_m) is defined by:

$$\phi_m = \frac{\sum_{k=1}^{K} |\phi_k - \phi_{pk}|}{K}, \tag{3.18}$$

$$\psi_m = \frac{\sum_{k=1}^{K} |\psi_k - \psi_{pk}|}{K}. \tag{3.19}$$

We use MAE instead of Euclidean distance between the GT and predicted orientation as in our work we found that sometimes a regressor is able to infer only one of the two angles correctly. In such a scenario, a Euclidean distance does not present accurate measure of performance. On the other hand, MAE provides a quantitative measure of regressor's performance independently for each orientation angle. Here, this measure is used evaluate the accuracy of the proposed methods. We use the MAE to define the CMAE as:

$$CMAE = \frac{\phi_m + \psi_m}{2}, \tag{3.20}$$

CMAE is particularly used for tuning different training parameters of the methods proposed in this work. Both MAE and CMAE are affected by outliers in predictions. Therefore, to overcome this affect, we additionally present and analyze GT versus predicted angles plots.

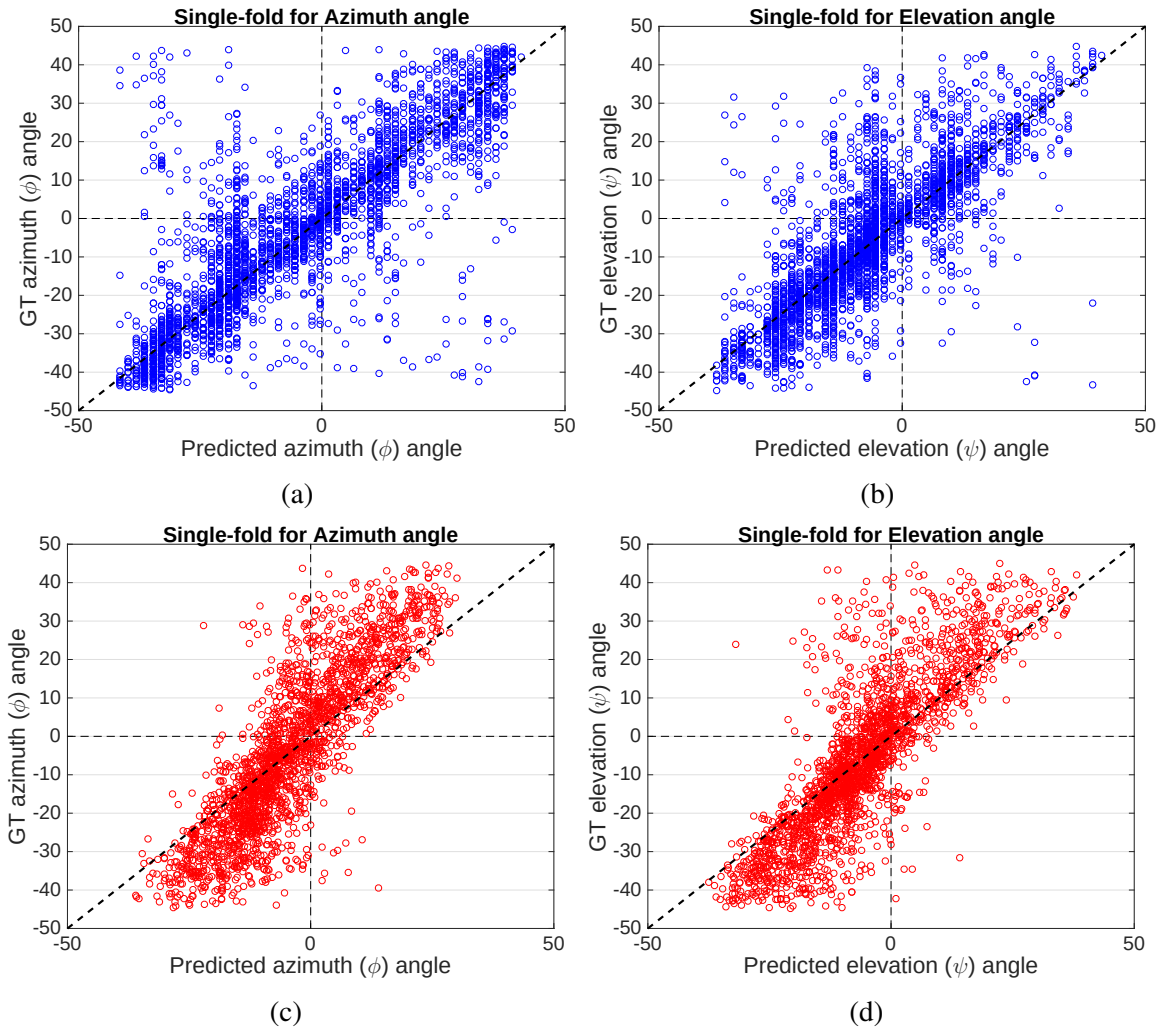


Fig. 3.13 GT versus predicted plots for (a)-(b) a model with outliers and (c)-(d) a model with bias in prediction.

3.7.2 Ground Truth versus Predicted Angles Plots

This dissertation uses GT versus predicted angle plots to visualize outliers and bias in a given regression model. Fig. 3.13 shows an example GT versus predicted angle plots for two different models showing both azimuth and elevation angle prediction. In these plots, the best model should have all points close to the $\phi_k = \phi_{pk}$ or $\psi_k = \psi_{pk}$ diagonal line. The results for the model shown in Fig. 3.13 (a)-(b) contain a number of outliers while the model shown in Fig. 3.13 (c)-(d) has model bias, which shifts some predictions by a certain value. It

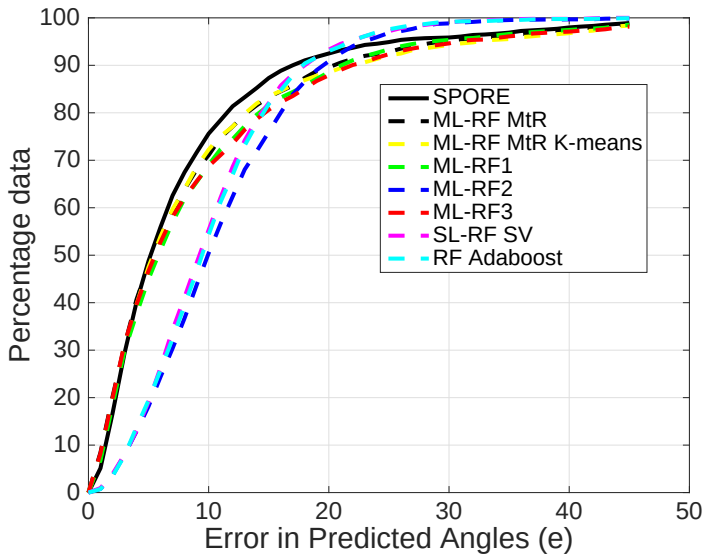


Fig. 3.14 Percentage data versus error plots comparing different models. A good performance is shown by a model that covers the highest percentage of data within a low error threshold, which in this case is achieved by SPORE.

should be noted that such plots add more insight into a given model's ability to learn, which is not fully depicted by MAE and CMAE.

3.7.3 Percentage Data versus Error Plots

Further insight into model performance is gained by constructing percentage data versus error plots. Similar plots have been used in literature for thorough validation of human pose as well as hand pose estimation [15, 16, 58]. These plots show percentage data that lies under an error threshold (e) as:

$$Percentage\ Data = \frac{1}{|\mathcal{S}|}\left(\sum_{k\in|\mathcal{S}|} \mathbb{1}_{((\phi_k<e)\ \wedge\ (\psi_k<e))}\right) \times 100, \tag{3.21}$$

where $\mathbb{1}_{((\phi_k<e)\ \wedge\ (\psi_k<e))}$ is an indicator function conditioned on $((\phi_k < e)\ \wedge\ (\psi_k < e))$.

Fig. 3.14 depicts an example percentage data versus error plot for a number of regression models. We note that the best performance is achieved by a regressor that covers most percentage data in under least error possible, which in this case is SPORE.

3.8 Summary

This chapter presented the background and terminology required to understand the work proposed in this dissertation. We gave an overview of a unified framework for hand orientation estimation. The different key components of the proposed framework were detailed, including the feature extraction and regressor training and prediction. Following this, we described the type of images used to collect the datasets in our work. A method was proposed for automatic and reliable annotation of GT hand orientations using depth images from a commodity depth camera. We also discuss the different collected datasets for our research. We concluded this chapter by describing the different error measures employed for evaluating the proposed methods along with what each error measure details about a model's performance.

Chapter 4

Hand Orientation Regression

This chapter presents the first realization of the unified framework introduced in Chapter 3. The proposed method learns a pair of Random Forest (RF) regressors for hand orientation inference. We present an overview of this method along with a flowchart. Following this, the dataset collection method is described along with the experimental evaluation and discussion of the results. We conclude this chapter by highlighting some limitations of the method, which are further addressed in the methods proposed in subsequent chapters.

4.1 Single-layered Random Forest for Hand Orientation Regression

We provide a realization of the unified framework to propose a single-layered RF orientation regression method. The flowchart in Fig. 4.1 shows the different steps in this proposed method, which has specific methods for each block in the unified framework. Both the training and prediction stages utilize the Contour Distance Feature (CDF) extraction method, described in Chapter 3, to extract features from hand silhouette images. For the training stage, Principal Component Analysis (PCA) of the training dataset is computed and the corresponding mean and eigenvectors are used to reduce the dimensionality of the CDF in both the training and prediction stages from 1000 down to 9 dimensions. Next, two single-

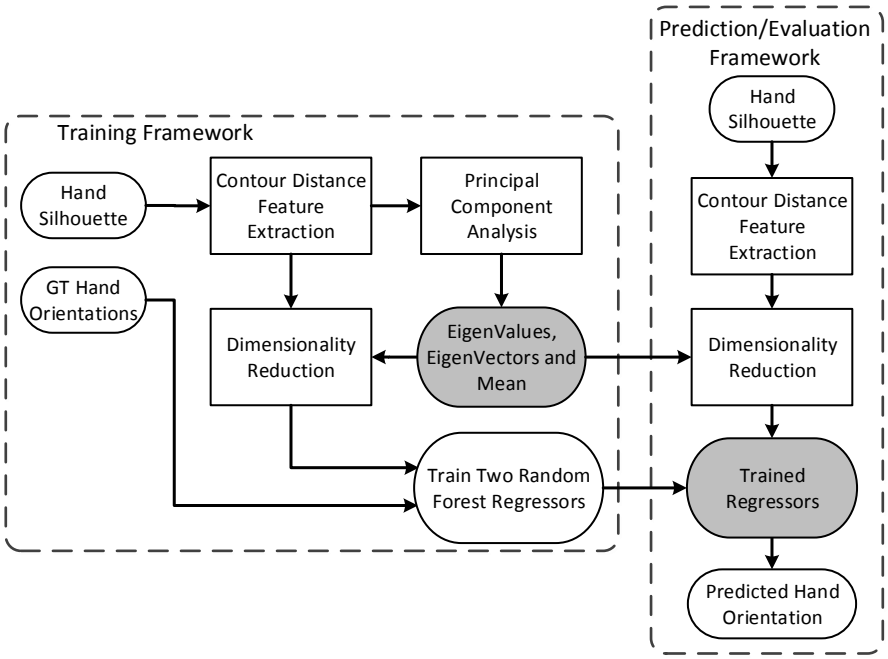


Fig. 4.1 Flowchart for the proposed single-layered Random Forest hand orientation regression. The trained regressor and extracted dimensionality reduction parameters are highlighted with a solid gray background.

variate RF regressors are independently trained to learn the mapping of the dimensionally reduced features onto each GT orientation angles. For the prediction stage, each trained regressor is used to infer the orientation angle using silhouette images only.

4.1.1 Dimensionality Reduction

The CDF extracted from the hand silhouettes have a large number of dimensions ($\Upsilon = 1000$). The work proposed in this chapter is our initial work that focuses on presenting a hand orientation regression method, where a dimensionally reduced feature space contributes to quick prototyping of the proposed framework. Thus, a post-processing step is introduced which is able to extract the prominent variations in the dataset by using PCA to project the feature vectors onto a reduced feature space [109]. We describe this post-processing step in further detail below.

The eigenvectors and eigenvalues of the corresponding CDF vectors are first extracted from the training data. The variations covered by each eigenvector is directly related to its

eigenvalue [110]. When sorted in decreasing order of variations, the first few eigenvectors capture the largest variations in the dataset. In our work, we use 90% energy defined by the most prominent eigenvalues to select the corresponding eigenvectors **e** for dimensionality reduction [111, 112]. This enables us to retain the primary variations in the dataset with reduced dimensionality, while leaving high frequency variations typically related to noise unmodeled. We assume that this unmodeled noise is represented by the left out eigenvectors corresponding to the lowest 10% energy. We note that similar methods have been previously used for image compression and noise reduction [113]. Projection of the feature vectors onto a reduced space can then be defined by:

$$\widehat{\mathbf{d}}_k = \mathbf{e}^T \left(\mathbf{d}_k - \boldsymbol{\mu}\right), \tag{4.1}$$

where $\boldsymbol{\mu}$ is the mean of all the samples and $\widehat{\mathbf{d}}_k$ is a set of dimensionally reduced feature vectors [114]. Fig. 4.2 shows the variations captured by the first six eigenvectors. The variations are shown as positive and negative modes defined as:

$$\mathbf{m}_{var} = \boldsymbol{\mu} + a\mathbf{e}_i, \tag{4.2}$$

where $-\sqrt{2} \leq a \leq +\sqrt{2}$ defines the amount of variations about the mean $\boldsymbol{\mu}$ and i is the index of the selected eigenvector. The effect of hand orientation variation on the CDF is seen in the first four variations shown in Fig. 4.2 (a)-(d). These correspond to changes in the height of peaks relative to the valleys depicting changes in elevation angles and the shift in the signal showing changes in azimuth angles. Fig. 4.2 (e)-(f) captures inter-person style variations where separation of different fingers, especially the thumb, pinky and ring finger, result in changes in CDF.

4.1.2 Hand Orientation Regression

The dimensionally reduced features $\widehat{\mathbf{d}}_k$ and GT orientation angles (ϕ_k, ψ_k) are used to train two RF regressors, one for each orientation angle. Both regressors are single-variate, regress-

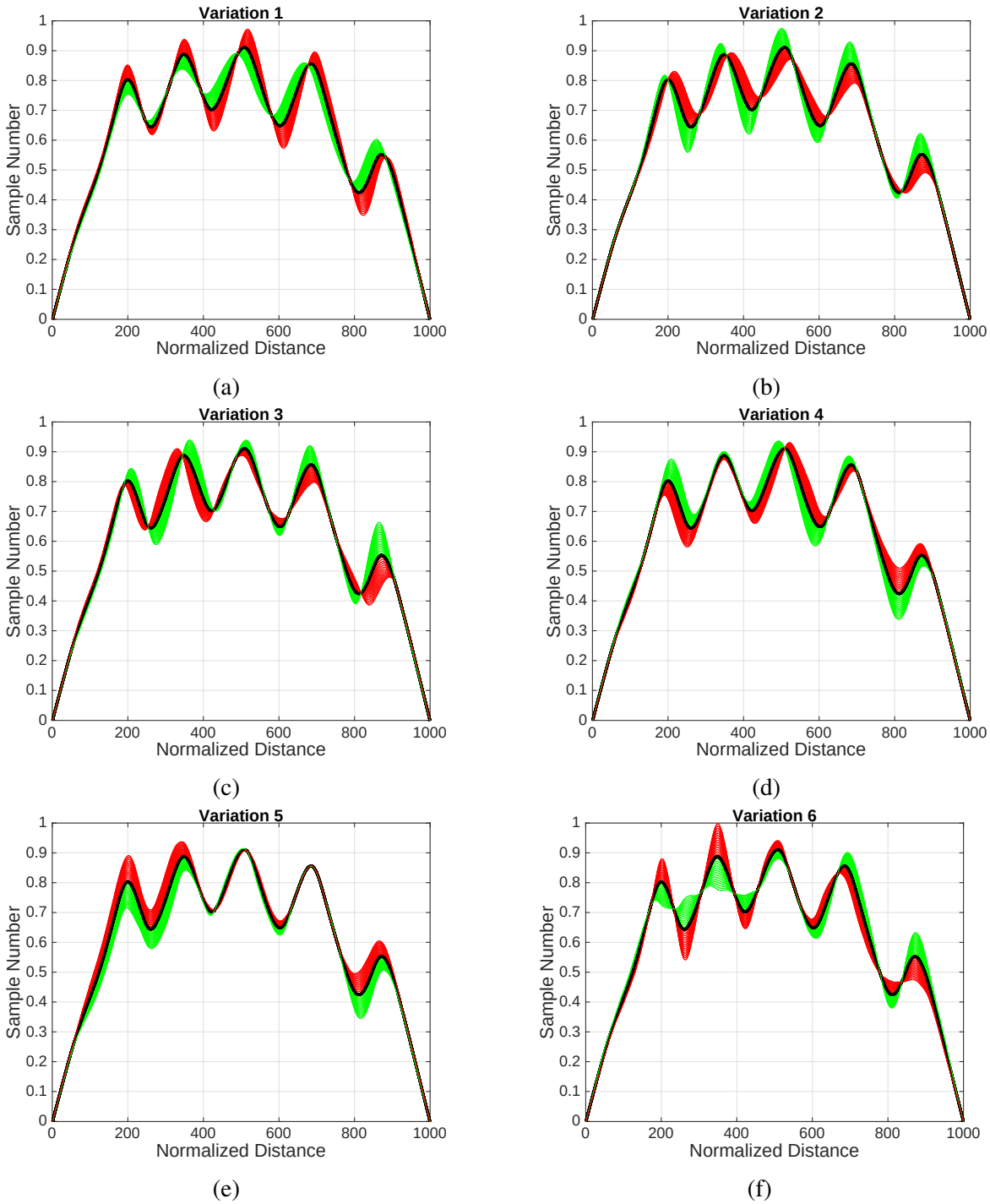


Fig. 4.2 The variations captured by first six eigenvectors used for dimensionality reduction. The mean of all samples is shown in black, whereas the positive modes and negative modes of variations are shown in green and red color, respectively.

ing against the orientation angles independently. In our experimental evaluation we use RF with 1000 trees and two features are sampled for splitting at each node. The RF learning algorithm described in Chapter 3 is used to train the regressors. During prediction, the dimensionality of an input CDF vector $\mathbf{d}_k$ is reduced using Equation 4.1. The dimensionally reduced feature vector $\widehat{\mathbf{d}}_k$ is then passed down in the two learned RF regressors where the leaf nodes infer the posterior probabilities $p(\phi|\widehat{\mathbf{d}}_k)$ and $p(\psi|\widehat{\mathbf{d}}_k)$. The final prediction is done by maximum likelihood estimate (MLE) defined as:

$$\phi^* = \arg\max_{\phi} p(\phi|\widehat{\mathbf{d}}_k). \tag{4.3}$$

$$\psi^* = \arg\max_{\psi} p(\psi|\widehat{\mathbf{d}}_k). \tag{4.4}$$

4.2 Evaluation of Single-layered Hand Orientation Regression

Evaluation of the proposed single-layered hand orientation regression framework is done using two different methods. A single-fold evaluation is done using 70% of the data for training while holding out 30% data for testing. Next, we perform a leave-one-out cross-validation, where in each trial we leave one participant's data out for training and use it to test the trained system. This latter validation method demonstrates how the system performs on unseen individuals, which quantifies the generalization capabilities of the proposed framework. In this section, we refer to our proposed single-layered Random Forest-based method with single-variate regressors as SL-RF SV. For comparison, all experiments are also repeated using a Neural Networks (NN) regressor with one hidden layer containing 1000 neurons. We note that at the time of this work, the author had limited understanding of NN. Hence, in this chapter we see a NN with a single hidden layer. We note that the Sections 5.2 and 6.2 in the next chapters provide an in-depth comparison of the proposed methods with relevant machine learning models. In our experiments, we varied the number of trees in

SL-RF SV and neurons in the NN regressors. However, changing these parameters did not significantly affect the output of our method. Therefore, we empirically fixed the number of trees and neurons to be 1000 for all the experiments. The data capture process and results are presented in subsequent sections below, which are then compared and discussed in Section 4.2.4.

4.2.1 Data Capture

The evaluation of the proposed SL-RF SV method is performed on *Dataset 1* which contains 1624 silhouette images and GT orientation collected from 13 participants. The choice of hand orientation variations used to record the dataset holds significance in depicting the contribution of the proposed framework. To generate this dataset we asked our participants to use an outstretched open hand pose throughout the data capture process. They were asked to rotate the hand back and forth, first along the azimuthal axis and then along the elevational axis only. Color images were segmented using the segmentation method in [90]. The segmented silhouette images and GT hand orientations were recorded while the participants performed these manipulations. As a result of different participants, the *Dataset 1* contains variations in hand size, shape and style of rotations for an open hand pose. This dataset only contains data from participants' right hand, however taking the advantage of mirror symmetry the same dataset can be reflected to generate images for the left hand.

The GT orientation angles are only used for the training step and are not part of the final prediction method, where only hand silhouettes from segmented color images are used. In *Dataset 1* both ϕ_k and ψ_k are limited from $-45°$ to $+45°$ to limit self-occlusions, particularly in the inter-finger regions, that occur beyond this range. It also provides a balance between the number of samples and the corresponding variations in the CDF. Furthermore, as shown in Fig. 4.3, the orientation angles further away from azimuth and elevation axes were discarded with a threshold of $\pm 10°$ for both ϕ_k and ψ_k. The selected orientation angles are useful for visual inspection of virtual objects where up/down and left/right movements are required. We present one such application in Section 4.3.

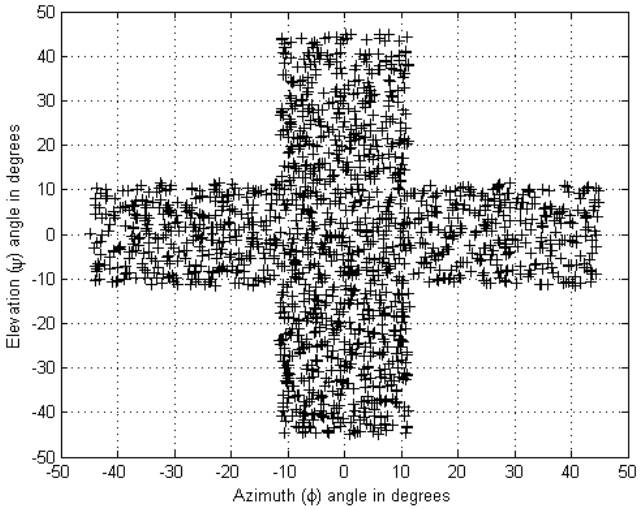


Fig. 4.3 Hand orientation space defined by the captured *Dataset 1* for evaluation of single-layered hand orientation regression method.

4.2.2 Single-fold Validation

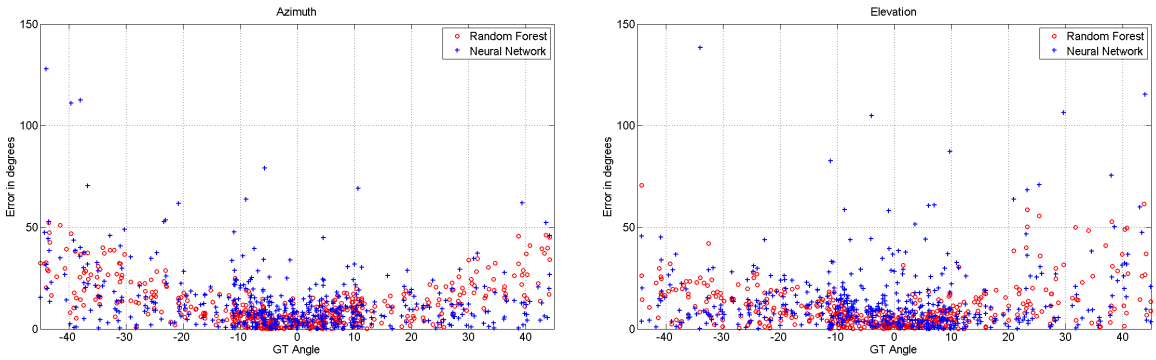
To evaluate the overall performance of the proposed regression method, we randomly divide *Dataset 1* into training and testing sets, with 70% and 30% of the total data, respectively. The system is then trained and evaluated using the corresponding sets of data.

The absolute prediction errors for this validation are calculated as:

$$\phi_{ak} = \left|\phi_k - \phi_{pk}\right|, \tag{4.5}$$

$$\psi_{ak} = \left|\psi_k - \psi_{pk}\right|, \tag{4.6}$$

where (ϕ_k, ψ_k) are the GT orientation angles, $\left(\phi_{pk}, \psi_{pk}\right)$ are the predicted angles and (ϕ_{ak}, ψ_{ak}) are the absolute prediction errors for the k^{th} sample. These errors are presented against GT orientation angles in Fig. 4.4. We also present plots of GT orientation angles against corresponding predicted angles in Fig. 4.5. For comparison, both these figures include results from SL-RF SV and NN. The Mean Absolute Error (MAE) (ϕ_m, ψ_m) for the single-fold validation is presented in Table 4.1. We also show success and failure cases for



(a) (b)

Fig. 4.4 Absolute prediction error (in degrees) illustrating errors (a) ϕ_e and (b) ψ_e in single-fold validation using SL-RF SV and NN regression techniques.

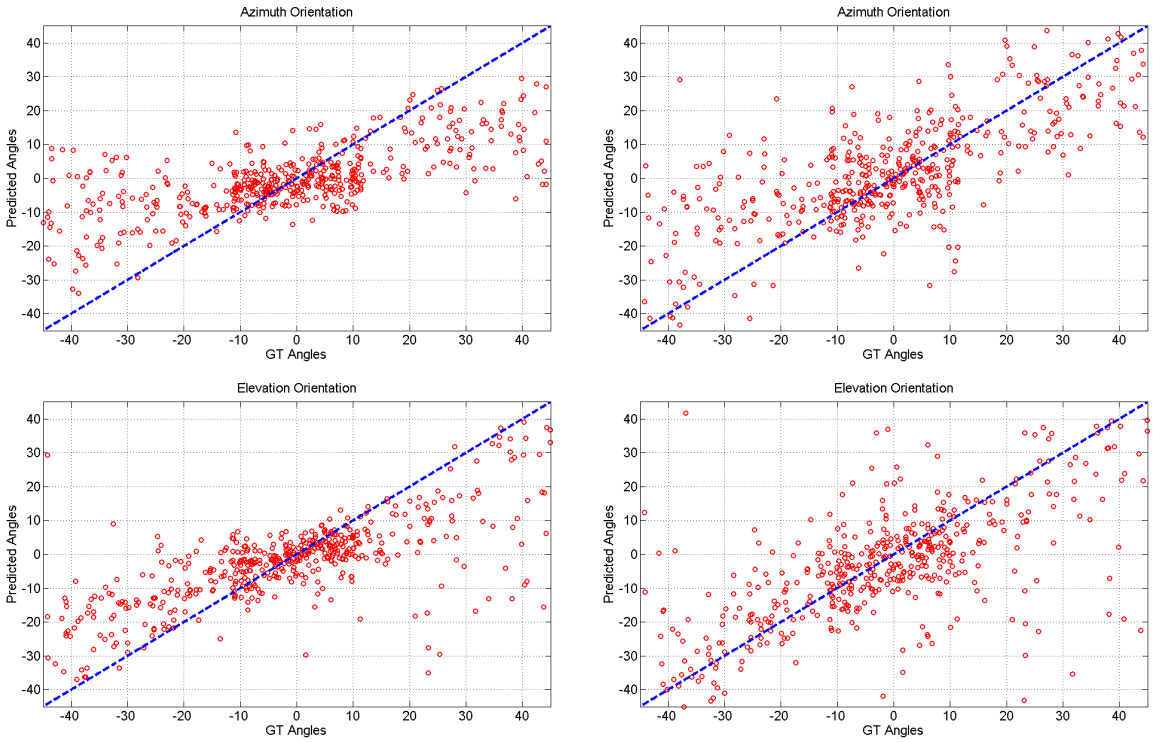
Evaluation method	**Regressor Used**	**Azimuth (ϕ_m)**	**Elevation (ψ_m)**
Single-Fold	SL-RF SV	**11.44** °	**9.57** °
	NN	15.31 °	14.19 °
Leave-one-out	SL-RF SV	**12.93** °	**12.61** °
	NN	20.14 °	18.85 °

Table 4.1 MAE in degrees for experimental evaluation of SL-RF SV hand orientation regression framework in Section 4.2.

the proposed SL-RF SV method in Fig. 4.7 and Fig. 4.8. Moreover, Fig. 4.9 shows failure cases that are affected by symmetry problem. The absence of depth information results in ambiguous predictions, where symmetrically opposite hand orientations are inferred. These experimental results indicate that the SL-RF SV regression method outperforms NN with an MAE of 11.44° for azimuth and 9.57° for elevation angles. We further discuss these results in Section 4.2.4.

4.2.3 Leave-one-out Cross-validation

We further evaluate our method against a scenario where in each trial, we leave one participant's data out from the training dataset. This left out data is then used for testing. This



(a) (b)

Fig. 4.5 GT versus predicted angle plots showing the accuracy of different regressors for predicted angles ϕ_{pk} and ψ_{pk} in single-fold validation. The predicted angles are illustrated using (a) SL-RF SV regressors with number of trees = 1000 and (b) NN regressor with one hidden layer containing 1000 neurons.

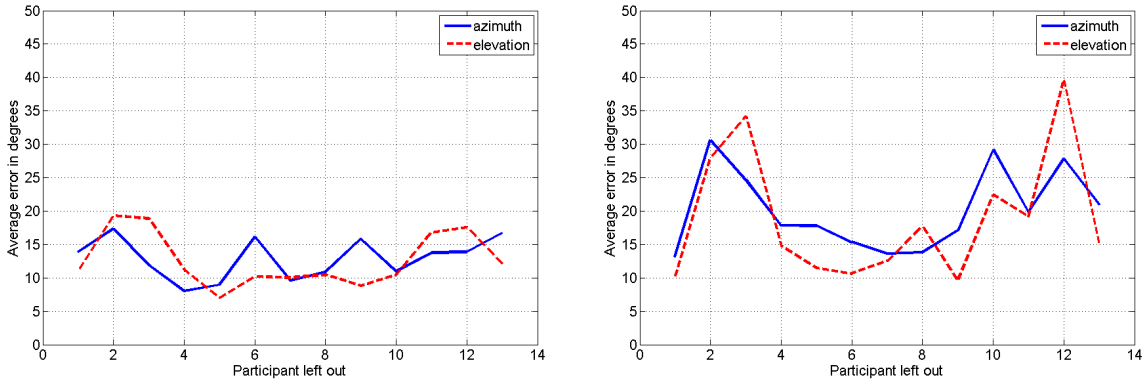
depicts a scenario where an unseen hand is used with our method. It is also able to evaluate the ability of the method to handle variations in hand shape, size and orientation without the need for an additional calibration step. The average prediction error for each participant using SL-RF SV and NN is presented in Fig. 4.6 (a) and (b) respectively, while Table 4.1 shows the results for average prediction error for all participants' cross-validation. These results indicate that the SL-RF SV method is able to generalize well from our hand orientation dataset with an MAE of 12.93° for azimuth and 12.61° for elevation angles. Further analysis and discussion of these results are presented in the following section.

4.2.4 Discussion

Experimental results show that the proposed method is able to learn the mapping from CDF to orientation angles. The method performs well when using SL-RF SV in both single-fold and leave-one-out cross-validation. The MAE for single-fold evaluation using SL-RF SV is close to 10° for both ϕ and ψ angles, as shown in Table 4.1. The average execution time of the proposed method for the given set of input silhouette images is found to be 16.93 ms per frame in a Matlab implementation on 3.2 GHz Core-i5 CPU.

Fig. 4.4 shows the absolute prediction errors against GT orientation angles for both SL-RF SV and NN regressors. It can be seen from this figure that SL-RF SV is able to model the underlying data well, with significantly fewer outliers as compared to the NN. Looking at the range -10° to $+10^\circ$, SL-RF SV is able to predict with better accuracy, while NN regressor has a significant number of outliers. As shown in Fig. 4.3, this range contains large variations in orientation and, consequently, the visible shape of the hand, which forms a benchmark for evaluating the generalization capabilities of the learned models. Prediction errors with the NN model in this range indicate that the learned model has memorized the training examples instead of generalizing from the data [115]. We note that if we eliminate the outliers then the NN performs better as shown in Fig. 4.4. Nevertheless, such outliers are a part of the system and result in greater overall average error for NN. On the contrary, results from SL-RF SV model depict its ability to apply learned representation to unseen examples. These differences in performance also arise due to two different learning methods. A NN model learns from a dataset by minimizing a global loss making it susceptible to overfitting, model bias and poor generalization. However, the trees in SL-RF SV are trained independently using a greedy algorithm that minimizes a local loss, enabling better generalization.

In Fig. 4.5, we establish the relationship between GT and predicted orientation angles to illustrate the performance of different regressors in single-fold validation. The diagonal line represents correct predictions. The closer the predicted data is packed around this diagonal, the better is the performance of the regressor. It can be seen from this figure that for both ϕ and ψ, SL-RF SV is able to perform better with fewer outliers. Additionally, from Fig. 4.5 (a) it can also be seen that the SL-RF SV predictions have a rotational bias as compared to NN

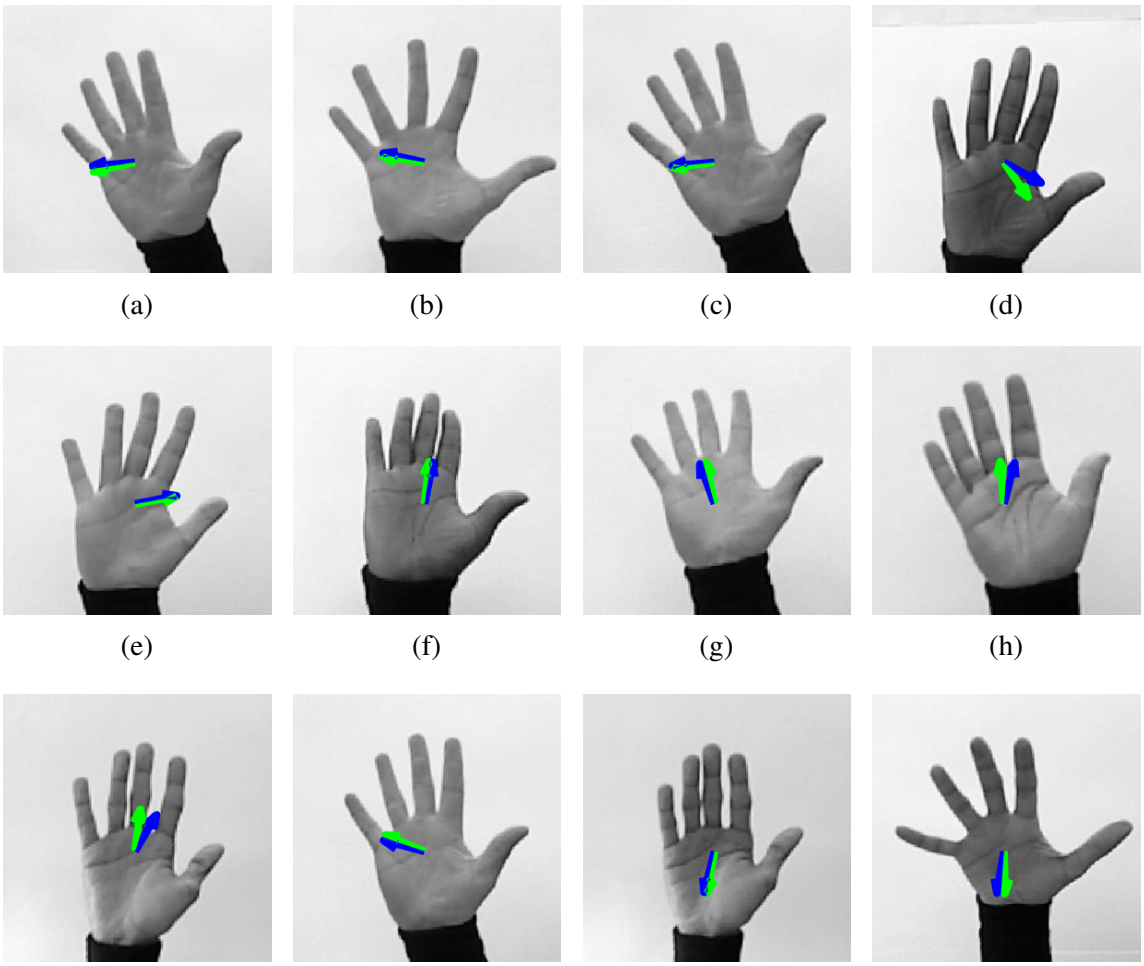


(a) (b)

Fig. 4.6 MAE in degrees for leave-one-out cross-validation of each participants' data using (a) SL-RF SV and (b) NN regression techniques.

predictions. This is an existing problem for SL-RF SV and a number of approaches have been previously proposed to debias the regression [116, 117]. In our problem, this bias is the result of the variations in *Dataset 1*. We address this in the subsequent chapters. Furthermore, the unguided dataset collection method used here presents a number of challenges, particularly in limiting the unwanted deviations from planar hand pose assumption. Moreover, as the method is unguided therefore only a small subset of orientation variations can be recorded in such method. This can be addressed by utilizing a feedback system where we can guide the participants to perform hand orientation variations that fully cover the orientation space. We present this improved data collection method in Chapter 5.

Leave-one-out cross-validation results show the compelling performance of the proposed SL-RF SV method for unseen hands. This evaluation method illustrates how well the system can generalize from a training data containing different variations in hand shape, size and style. Comparing the average prediction errors for leave-one-out cross-validation with single-fold validation in Table 4.1, there is a significant decrease in the performance of the NN regressors. This highlights the inability of the NN to generalize for unseen data. Fig. 4.6 further validates these results for each individual participant. In this validation SL-RF SV produces relatively lesser error, which indicates its ability to generalize better from the inter-person variations.



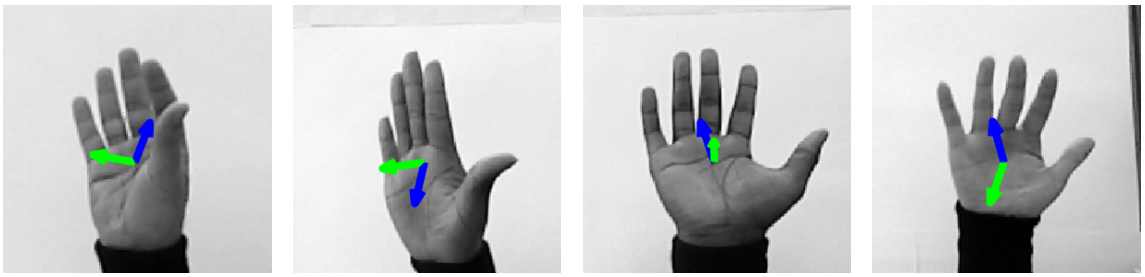
(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

	Absolute Error			**Absolute Error**	
	ϕ_{ak}	ψ_{ak}		ϕ_{ak}	ψ_{ak}
(a)	0.89°	3.61°	(g)	5.17°	1.35°
(b)	4.33°	0.39°	(h)	9.08°	0.04°
(c)	1.18°	0.52°	(i)	7.83°	1.31°
(d)	0.81°	8.69°	(j)	7.12°	0.20°
(e)	1.17°	0.34°	(k)	1.07°	0.86°
(f)	3.18°	0.88°	(l)	7.18°	1.22°

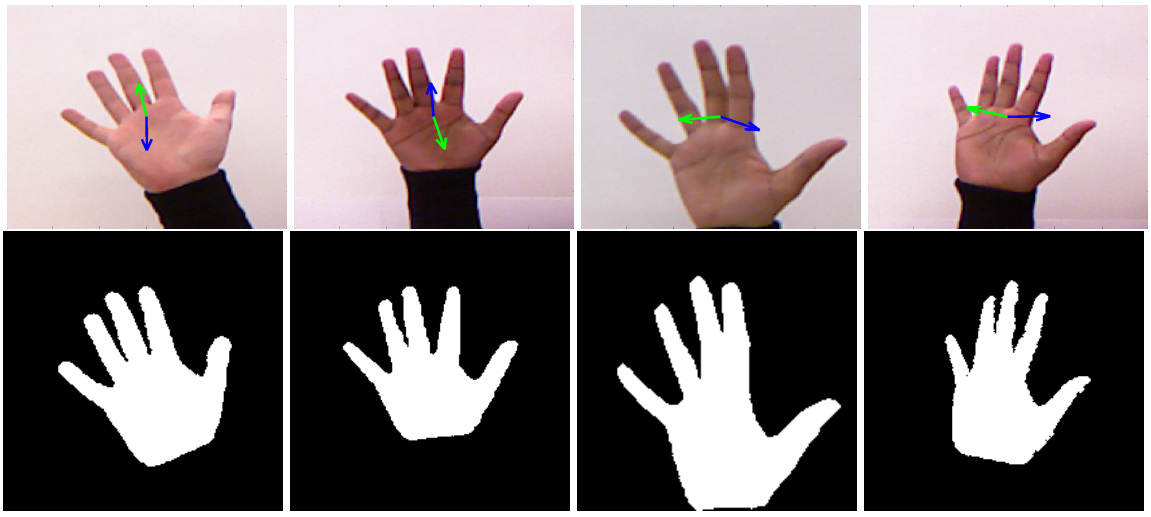
Fig. 4.7 Success cases for our proposed method. The GT normal vectors are superimposed on each image using green arrows, whereas predicted normal vectors are shown using blue arrows. Absolute prediction errors for each case are shown in the table above.



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

	Predicted		Ground Truth		Absolute Error	
	ϕ_{pk}	ψ_{pk}	ϕ_k	ψ_k	ϕ_{ak}	ψ_{ak}
(a)	2.96°	−7.90°	−43.46°	−7.26°	46.42°	0.64°
(b)	−0.54°	2.29°	−43.29°	5.85°	42.75°	3.56°
(c)	−3.86°	−11.16°	−1.15°	−41.17°	2.71°	30.01°
(d)	−4.88°	−17.85°	−7.00°	20.97°	2.12°	38.82°

Fig. 4.8 Failure cases for our proposed method. The GT normal vectors are superimposed on each image using green arrows, whereas predicted normal vectors are show using blue arrows. Error for each case is presented separately in the table above.



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

Fig. 4.9 Symmetry problem in silhouette images shows color images of the hand in the top row and the segmented silhouette images in the bottom row. The absence of depth information in silhouette images results in similar silhouettes for symmetrically opposite hand orientations. The images in (a)-(b) and (c)-(d) are from symmetrically opposite orientations of the hand.

As discussed above the proposed SL-RF SV hand orientation regression method performs well to recover 3D hand orientation in the presence of a number of underlying variations in hand shape, size and style. In Fig. 4.7 we present different success cases using SL-RF SV in single-fold validation. It also shows a table with the absolute prediction errors for each of these success cases. The variations in *Dataset 1* and the capability of our method are clear from these results.

While our method performs well for most of the cases, it fails in some cases. Fig. 4.8 shows some of the cases where our method fails along with a table showing the corresponding error. These failure cases can easily be identified as outliers in the dataset as they do not have outstretched hand pose. In Fig. 4.8 (a), (c) and (d) the hand does not follow the planar surface assumption which directly affects the calculation of GT orientation angles, whereas in Fig. 4.8 (b) the fingers are placed too close together making it impossible to extract a CDF vector that corresponds to the ones in the training dataset. Furthermore, by analyzing the absolute prediction errors for each failure case in the table, it can be seen that our method only fails for the orientation where these assumptions fail. Since our method is regressing both orientation angles independently, therefore, even in these failure cases, the unaffected angle is predicted with good accuracy. The assumption that both orientation angles vary independently may be true for the given *Dataset 1*, where the orientation space is only defined along the orientation axes, however, the variations in these angles can be correlated for a fully covered orientation space. Replacing the single-variate regressor with a multi-variate regressor will use the orientation dependence of both angles to better model the orientation space. We discuss this in further detail in subsequent chapters. We also show some failure cases, where symmetrically opposite orientations are inferred (shown in Fig. 4.9). This is due to the absence of depth information, where multiple symmetrically opposite orientations result in similar silhouette images. Comparing the silhouette images in Fig. 4.9, we note that these cases look similar for symmetrically opposite orientations.

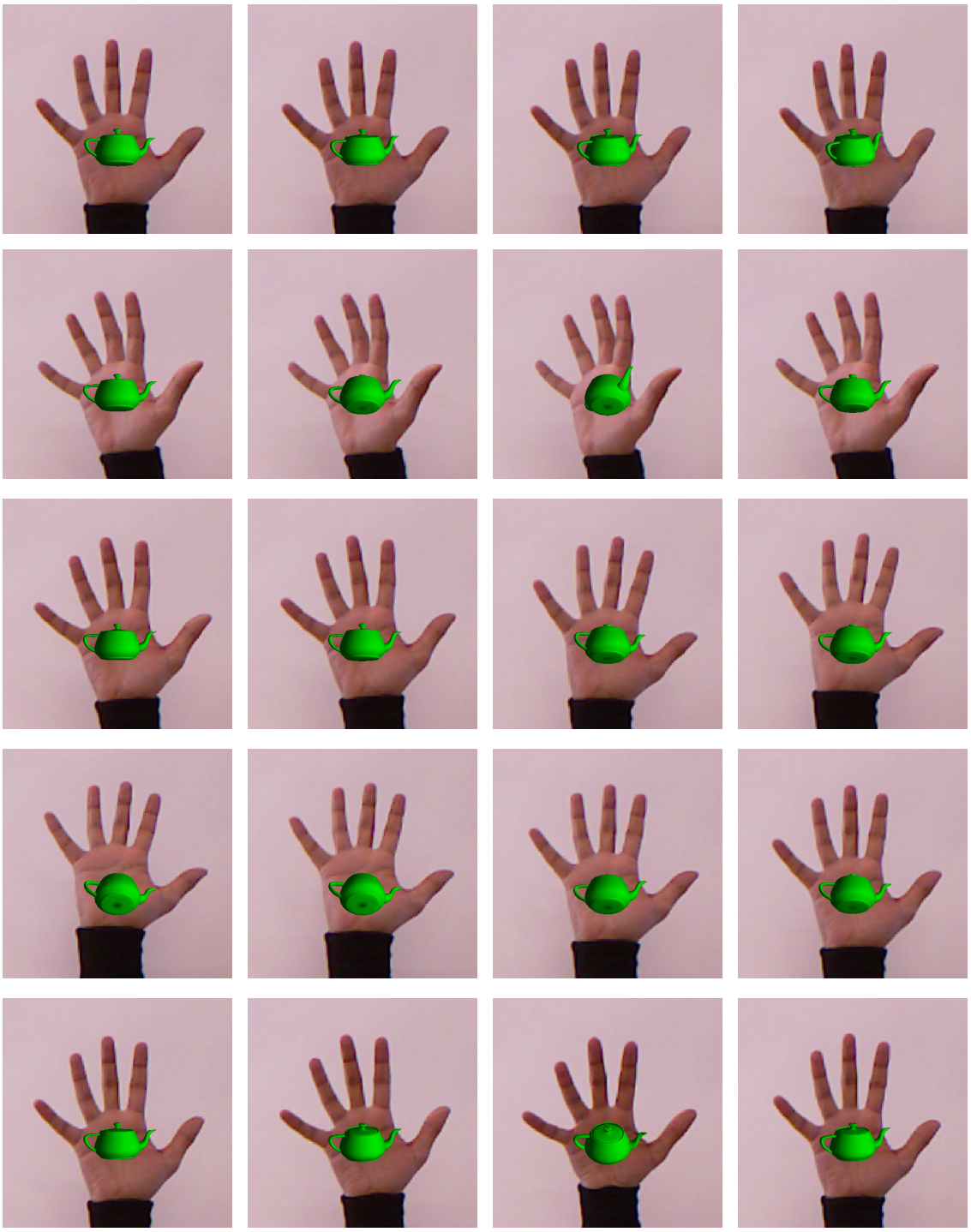


Fig. 4.10 Application of the proposed approach to Augmented Reality for visual inspection of virtual objects. In this example, the orientations of a virtual teapot are controlled with the hand orientations extracted using the proposed method. The sequence of hand orientation starts from the top left and ends at the bottom right image.

4.3 Application to Augmented Reality

The proposed method can be applied to a number of different applications. In our work, we present an Augmented Reality-based application for visual interaction with virtual objects (shown in Fig. 4.10). In this application, the virtual object is overlaid on an augmented layer. Orientation changes from the hand movements are captured using our method and the corresponding orientation transformations are applied to the augmented object. This kind of visual inspection of virtual objects is useful in scenarios where users do not have access to the actual object, however, they want to view it from different perspectives.

Another scenario where the hand orientation could be used is in computer games. We present a simple ball bouncing game where an incoming ball is directed into different target directions using just the orientation of the hand. Fig. 4.11 shows screenshots of this game. Similar games can be implemented that utilize the orientation of the hand to solve certain objectives. As the control is directly coming from the user's hand, the game gives a natural interaction experience.

Although our work does not study any potential improvements in interaction when using hand orientation for the proposed application scenarios, we do have some interesting insights from the application testing. In our testing, we noticed that each new participant quickly grasped the idea of orientation-based object manipulation. We note that utilizing hand orientation in Augmented Reality can enable users to naturally interact with a virtual object, as such manipulations are part of our daily interaction with other physical objects. Further studies could explore an in-depth analysis of our proposed interaction scenario for improving human-computer interaction.

4.4 Summary

This chapter proposed a single-layered Random Forest regressor for learning the mapping of dimensionally reduced CDF onto hand orientation angles. The proposed model was trained and evaluated on a dataset of hand silhouette images and orientation angles. The system performed well with an MAE of 10° for single-fold validation and 12° for leave-one-out

Fig. 4.11 Application of the proposed approach to an Augmented Reality game. The objective of the game is to deflect the incoming balls to target directions using the orientation of the hand. The sequence of hand orientations starts from the top left and ends at the bottom right image.

cross-validation. The proposed method was shown useful for an Augmented Reality-based visual inspection application of virtual objects. A ball bouncing game was also shown as a possible application of the inferred hand orientations. The results indicate a number of areas that require further research. Particularly, the unguided data collection method proved limited as it enabled collection of only a small dataset with limited variations in hand orientations. Furthermore, it was found that the lack of feedback during capture resulted in a number of complex inter-person hand shape variations in which the user deviated from the planar hand

pose assumption. SL-RF SV was affected by the model bias which indicates that the model used was unable to fully learn from the variations in the dataset. We address these limitations in the next chapter where we propose a multi-layered Random Forest method that learns from specific variations within the dataset.

Chapter 5

MtR: Learning Marginalization Through Regression for Hand Orientation Inference

This chapter addresses the challenges faced by the single-layered hand orientation regression method proposed in Chapter 4. The model proposed in Chapter 4 was unable to fully learn the mapping of Contour Distance Features (CDF) onto hand orientation angles, where a model bias was noticed. Another problem faced by the method proposed in Chapter 4 was the use of unguided dataset collection method. This limited the amount of reliable data that could be captured, where the participants relied on the description of hand orientation movements.

Existing work has utilized multi-layered Random Forest (ML-RF) for dividing complex learning tasks and learning using multiple expert regressors to achieve better accuracy than a single-layered model [17, 21]. Fig. 5.1 explains this concept with a simplified abstract example. It shows that when the underlying problem contains a number of variations, then a single model is unable to learn from all the variations. Fig. 5.1 (b) shows how a single model *tries* to fit in to the data, resulting in a model bias. This problem can be solved by introducing a number of expert regressors where each regressor can be trained on subsets of the data to learn specific variations (shown in Fig. 5.1 (c)). We visualize the variations

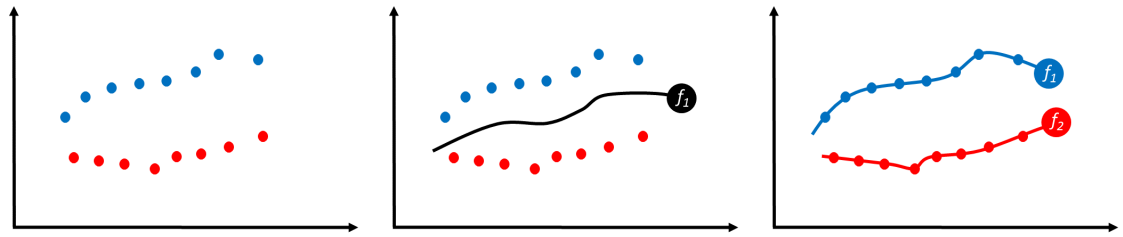
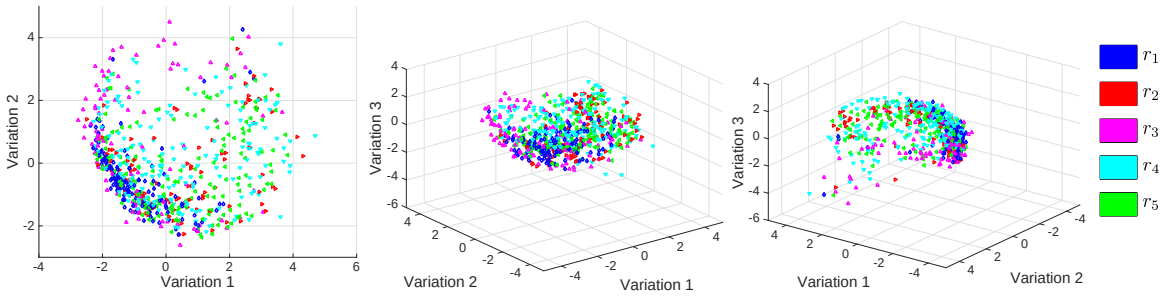


Fig. 5.1 Learning from a number of variations within the data shows (a) input data with two types of variations, (b) a single learned model is shown in black and (c) two learned models trained for specific variations. In scenarios where the dataset contains a number of variations, a single model is unable to fully learn the mapping resulting in model bias as shown in (b). However, when the underlying problem is divided into subproblems, a set of models can fully learn from the variations within a dataset.

contained in our *Dataset 2* in Fig. 5.2, where projections of CDF vectors onto three dominant variations defined by eigenvectors are shown. Colors from latent variable definition in Section 5.1.1 are shown to represent different orientations of the hand. It can be seen that the hand shape in different orientations produces different variations resulting in a complex mapping. Therefore, a single regression model proves insufficient for learning the mapping of CDF vectors onto the corresponding hand orientation angles. Hence, we utilize multiple expert regressors, where each regressor learns from a specific variation in the dataset. However, such a model choice poses a number of challenging questions. These include determining the best way to divide the dataset into subsets and determining, during online prediction, which expert regressor(s) are capable of making the most accurate prediction for a given input feature vector.

In this chapter, we propose a multi-layered Random Forest with Marginalization through Regression (ML-RF MtR) method to address hand orientation regression [30]. ML-RF MtR presents a method for dividing dataset using a latent variable space, where each latent variable forms a subset. Expert regressors are trained on these subsets. During online prediction, each expert regressor gives a posterior probability for a given latent variable. Our main contribution comes from the regression-based marginalization of these posterior probabilities. Furthermore, this chapter also presents a data collection method, where the participants are provided with feedback which acts as a guidance for collecting similar orientations



(a) (b) (c)

Fig. 5.2 Variations in the *Dataset 2* used in this chapter. The most dominant variations in the dataset are shown by projecting CDF vectors using the first three eigenvectors of *Dataset 2*. The five different coarse orientations, defined by latent variable space proposed in this chapter, are shown with same colors as in Fig. 5.5. The variations in the dataset are complex and in some places not easily separable by a single model. This motivates the need for a method that depends on multiple expert regressors, each learning from a subset of variations in the data.

from a number of different users. We also show the application of our proposed method to augmented character's manipulation in an egocentric video scenario.

5.1 Marginalization through Regression for Multi-layered Random Forest

Inspired by existing work in [17, 21, 34, 66], where complex problems have been divided and solved by a number of expert regressors trained on simpler subsets of the data, we present a novel ML-RF MtR method that is capable of learning from a number of variations in our hand orientation dataset [30]. The proposed model utilizes multi-variate Random Forest (RF) regressors, that are able to simultaneously learn the mapping of CDF onto both hand orientation angles. Fig. 5.3 shows the proposed ML-RF MtR model architecture, which is split into two layers, namely, marginalization weights and expert regression layer. The proposed method is divided into training and prediction frameworks as shown in Fig. 5.4. The training framework uses a latent variable space along with the GT hand orientation angles to split the input data into a number of subsets. Expert regressors are then trained on the defined subsets of the data. The posterior probabilities corresponding to each sample in the training

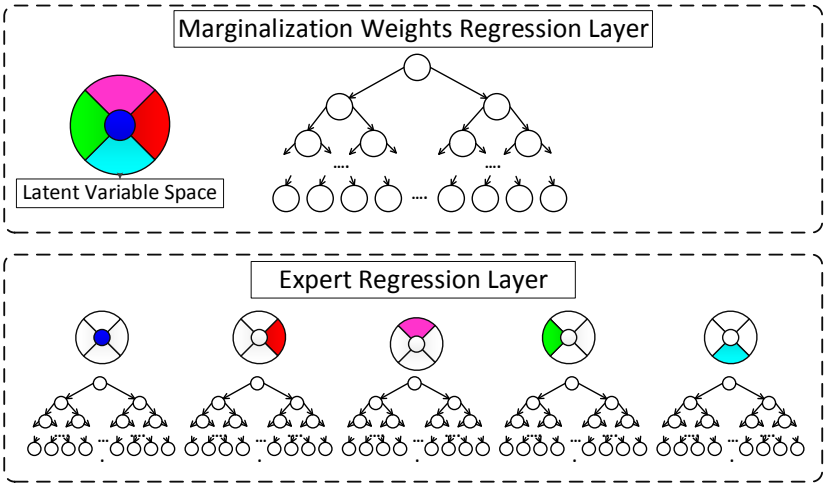


Fig. 5.3 The proposed multi-layered marginalization through regression method utilizes marginalization weights regressor in the first layer to infer the weights for marginalizing posterior probabilities from each expert regressor in the second layer.

set are acquired from each of these regressors. Our main contribution comes from the use of a marginalization weights regressor that learns the mapping of CDF to marginalization weights. We derive and apply a Kullback-Leibler divergence-based optimization technique that estimates the marginalization weights for training data. The online prediction for a given CDF utilizes marginalization weights regressor to predict marginalization weights. Posterior probabilities from each expert regressor are then extracted. These probabilities are marginalized using the inferred marginalization weights. The resulting marginalized probability is used to infer the hand orientation angles. We further describe the proposed ML-RF MtR method in the following sections.

5.1.1 Latent Variable Generation

We define a latent variable space to divide our training dataset into subsets. This space is based on the simple observation that the hand orientation can be broadly categorized with respect to the camera as being: (i) fronto-parallel, or facing (ii) right, (iii) left, (iv) upwards or (v) downwards, which also corresponds to the maximum distinctive hand shape variations.

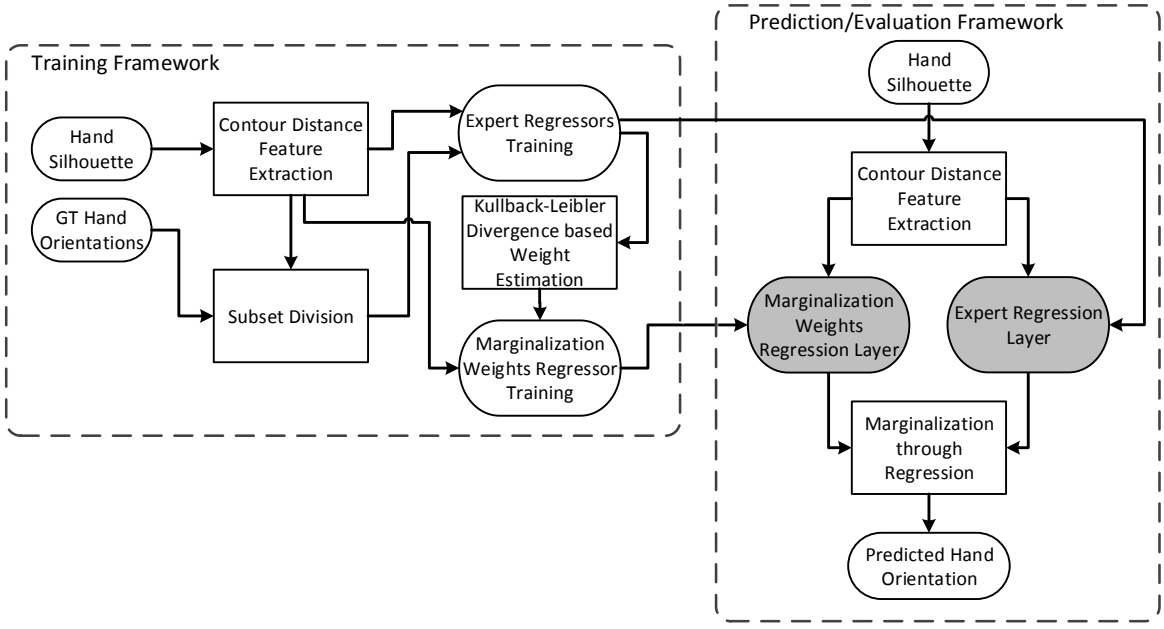


Fig. 5.4 Flowchart for training and evaluation of the proposed marginalization through regression method for hand orientation inference. The trained regressors are highlighted with solid gray background.

Each set of GT orientation angles (ϕ_k, ψ_k) are first transformed into polar coordinates (γ_k, φ_k) and are then used to generate latent variables for dividing the target space into five different regions as:

$$r_n(k) = \begin{cases} r_1 & \text{if } \gamma_k \leq \alpha^\circ, \\ r_2 & \text{if } \gamma_k > \alpha^\circ : \varphi_k \in (0^\circ - \beta^\circ, 90^\circ - \beta^\circ], \\ r_3 & \text{if } \gamma_k > \alpha^\circ : \varphi_k \in (90^\circ - \beta^\circ, 180^\circ - \beta^\circ], \\ r_4 & \text{if } \gamma_k > \alpha^\circ : \varphi_k \in (180^\circ - \beta^\circ, 270^\circ - \beta^\circ], \\ r_5 & \text{if } \gamma_k > \alpha^\circ : \varphi_k \in (270^\circ - \beta^\circ, 360^\circ - \beta^\circ], \end{cases} \tag{5.1}$$

where α and β are adjustable parameters defining the radius of the central region and the offset for the latent variable space, respectively, and $r_n \in \{r_1, r_2, r_3, r_4, r_5\}$ are the latent variable labels dividing the dataset for ML-RF MtR regression (shown in Fig. 5.5).

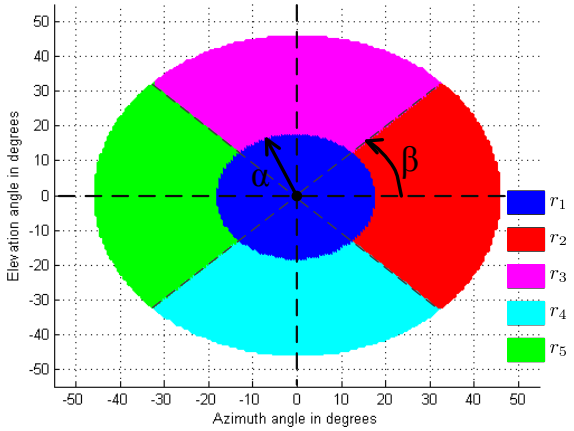


Fig. 5.5 Latent variable space showing the different latent variables in different colors. The parameters α and β define the radius of the central region and the offset for the latent variable space, respectively.

5.1.2 Expert Regression Layer

A set of multi-variate RF regressors are trained on the subsets of the data to learn the mapping of hand CDF onto orientation angles. These regressors form the expert regression layer. Each regressor in this layer is trained using a subset of the data defined by latent variable $r_n \in \{r_1, r_2, r_3, r_4, r_5\}$. Training is achieved using a standard RF regression method described in Section 3.4.1. Given an input CDF vector $\mathbf{d}_k$, the posterior probabilities for orientation angles (ϕ, ψ) for a given latent variable r_n are given by this layer as:

$$p(\phi, \psi | r_n, \mathbf{d}_k) = \frac{1}{T} \sum_t p_t(\phi, \psi | r_n, \mathbf{d}_k), \tag{5.2}$$

where $p_t(\phi, \psi | r_n, \mathbf{d}_k)$ is the posterior probability from leaf node of tree t and T is the total number of trees in a given RF model.

In our framework we are interested in finding a marginalized posterior probability $p(\phi, \psi | \mathbf{d}_k)$, which can be utilized in maximum likelihood estimate (MLE) to make predictions. A common method for acquiring the marginalized posterior probability, as explained

in Section 3.2.2, is given by:

$$p(\phi, \psi|\mathbf{d}_k) = \sum_n p(\phi, \psi|r_n, \mathbf{d}_k). \tag{5.3}$$

However, as our expert regressors are trained on different subsets of the data, the posterior probability $p(\phi, \psi|r_n, \mathbf{d}_k)$ is influenced differently by the possible values of latent variable r_n. Moreover, there exist some predictions, especially for the samples that lie on the boundary of multiple latent variables, where posterior probabilities from only a few expert regressors contribute towards accurate predictions. To accommodate this, marginalization weights ω_{nk} are introduced as:

$$p(\phi, \psi|\mathbf{d}_k) = \sum_n p(\phi, \psi|r_n, \mathbf{d}_k)\, \omega_{nk}, \tag{5.4}$$

where the marginalization weights ω_{nk} correspond to each latent variable such that $\sum_n \omega_{nk} = 1$. The conditional probability $p(\phi, \psi|\mathbf{d}_k)$ can be used to make the overall regressor prediction defined by MLE estimate as:

$$(\phi^*, \psi^*) = \underset{(\phi,\psi)}{\arg\max}\, p(\phi, \psi|\mathbf{d}_k). \tag{5.5}$$

Next, we propose a method for estimating these weights given a learned expert regression layer.

5.1.3 Marginalization through Regression

To learn the marginalization weights regressor, we must first determine the GT weights ω_{nk}, which form the regression target. This regressor, highlighted as a gray box in flowchart in Fig. 5.4, learns the mapping of CDF $\mathbf{d}_k$ onto the multi-variate marginalization weights vector ω_k. We formulate the GT probability for the training samples using the GT orientation angles (ϕ_k, ψ_k) in a multi-variate normal distribution as:

$$p(\phi_k, \psi_k) = \mathcal{N}((\phi_k, \psi_k), \Sigma), \tag{5.6}$$

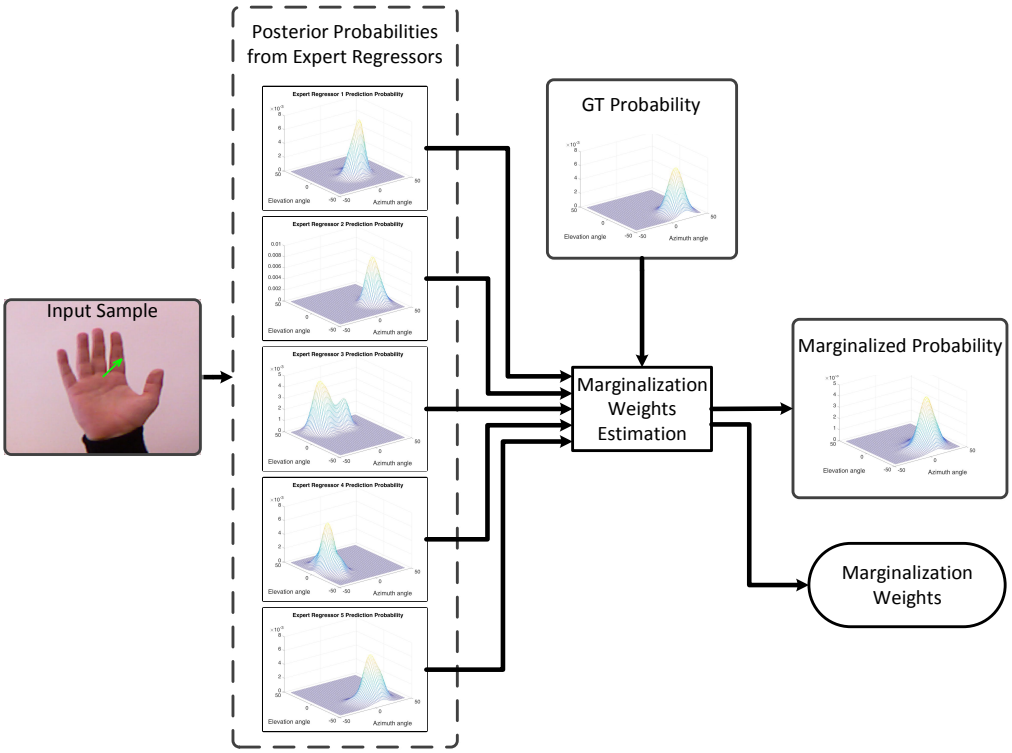


Fig. 5.6 Marginalization weights estimation using training data. A training sample is used to get posterior probabilities from each expert regressor. These probabilities are then used along with the GT probability in Equation 5.7 to estimate marginalization weights and the corresponding marginalized probability.

where Σ is the covariance that can be adjusted to control the spread of $p(\phi_k, \psi_k)$.

Given the GT probability $p(\phi_k, \psi_k)$ and the corresponding posterior probabilities $p(\phi, \psi | r_n, \mathbf{d}_k)$, we propose a novel optimization method, where the marginalization error is based on Kullback-Leibler divergence [118]. This error is optimized to estimate the GT marginalization weights ω_{nk} for all latent variables $r_n \in \{r_1, r_2, r_3, r_4, r_5\}$. Fig. 5.6 shows this marginalization weights estimation framework. We define this error as:

$$E = \iint p(\phi_k, \psi_k) \log \frac{p(\phi_k, \psi_k)}{p(\phi, \psi | \mathbf{d}_k)} d\phi d\psi. \tag{5.7}$$

Derivation

Here we present the derivation of partial derivatives from Equation 5.7 that can be used to determine the optimal weights ω_{nk}. We can rewrite Equation 5.7 as:

$$\begin{aligned} E &= \iint p(\phi_k, \psi_k) \log \frac{p(\phi_k, \psi_k)}{p(\phi, \psi | \mathbf{d}_k)} d\phi d\psi, \\ &= \iint p(\phi_k, \psi_k) [\log\ p(\phi_k, \psi_k) \\ &\quad - \log[\sum_n p(\phi, \psi | r_n, \mathbf{d}_k)\, \omega_{nk}]] d\phi d\psi. \end{aligned} \tag{5.8}$$

The partial derivative w.r.t ω_{nk} can then be defined as:

$$\frac{\partial E}{\partial \omega_n} = -\iint \frac{p(\phi_k, \psi_k)\, p(\phi, \psi | r_n, \mathbf{d}_k)}{\sum_n p(\phi, \psi | r_n, \mathbf{d}_k)\, \omega_{nk}} d\phi d\psi. \tag{5.9}$$

Optimization

We use a standard gradient descent with,

$$\nabla E = \left[\frac{\partial E}{\partial \omega_{1k}}, \frac{\partial E}{\partial \omega_{2k}}, \frac{\partial E}{\partial \omega_{3k}}, \frac{\partial E}{\partial \omega_{4k}}, \frac{\partial E}{\partial \omega_{5k}} \right], \tag{5.10}$$

for which the optimization is iteratively evolved for a solution given by:

$$\omega_{nk}^{\upsilon+1} = \omega_{nk}^{\upsilon} - \lambda \nabla E^{\upsilon}, \tag{5.11}$$

where λ is the step size along the negative gradient direction and υ is the iteration number.

Marginalization Weights Regressor

We use a multi-variate RF regressor to learn the mapping of CDF $\mathbf{d}_k$ to marginalization weights ω_k. As shown in flowchart in Fig. 5.4, this regressor is used during prediction to infer marginalization weights ω_k for marginalizing the posterior probabilities $p(\phi, \psi | r_n, \mathbf{d}_k)$ from each expert regressors using Equation 5.4.

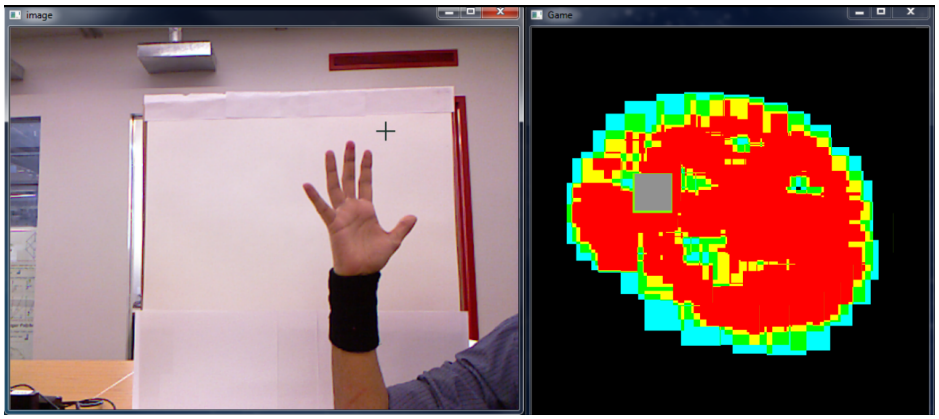


Fig. 5.7 Painting game application for capturing hand orientation *Dataset 2*, *3* and *4*, shows the captured hand image and the corresponding painting canvas giving feedback to the participant.

5.1.4 Data Capture

The proposed method is tested on a dataset of real color images and GT orientation angles captured using a Microsoft Kinect v1 sensor. To capture *Dataset 2*, *3* and *4* (listed in Table 3.1), we have developed a simple painting game application (shown in Fig. 5.7). The objective of the game is to paint on a digital canvas using hand orientation movements originating from flexion and extension of the wrist and supination and pronation of the forearm, as shown in Fig. 1.1. The canvas is designed to directly represent the hand orientation space in which different color hues represent the number of samples collected for a particular region. These include, black color for regions where no samples have been collected, whereas the painting color changes from green, representing the lowest number of samples, to red which represents regions that have been densely sampled. The objective of the game is to paint as much red as possible using a painting cursor shown in gray color. Such a game encourages participants to move hands in all possible orientations inside the defined orientation space. Thus, achieving consistency between orientation angles and hand shape for different users in the dataset.

For the data collection process, we asked our participants to paint a set of digital canvases using a planar hand pose. Real-time feedback of the canvas was provided to assist in the movement of the painting cursor. Hand orientation data was captured, using the Microsoft

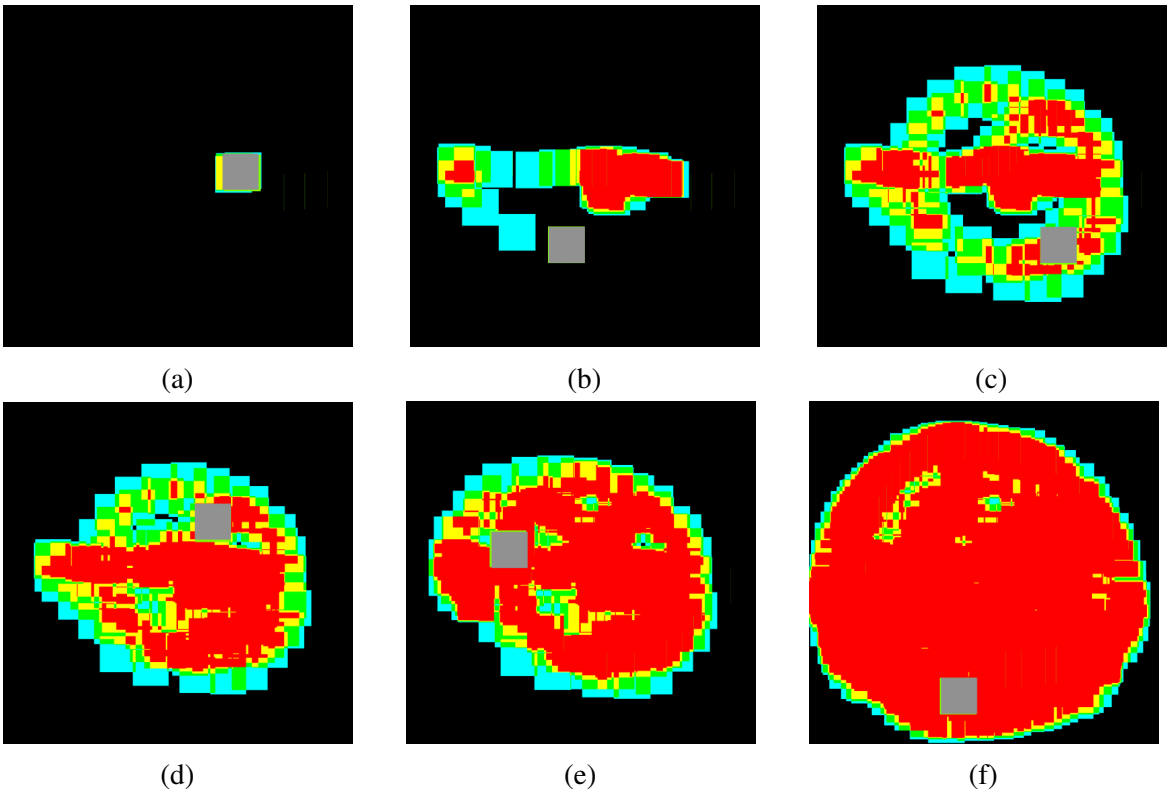


Fig. 5.8 Digital painting canvas for the data capture application. The canvas directly represents the hand orientation space with colors indicating the number of samples collected in the specific regions of the orientation space. A black color indicates the absence of the samples, while green depicts a low number of samples and red corresponds to densely sampled regions. The painting cursor is shown in gray. The data capture process progresses in time from (a)-(f), where (f) shows a densely sampled orientation space.

Kinect v1 sensor, while the participants completed the painting task. Fig. 5.7 shows this data capture setup with a participant's hand. The transformation of canvas through time is shown in Fig. 5.8. Samples from this painting application were collected by segmenting the hand region using the segmentation method from [90] and performing RANSAC-based GT orientation annotation using the depth image as described in the Chapter 3.

5.2 Experimental Evaluation

We evaluate the proposed ML-RF MtR method on *Dataset 2* that contains 7059 samples collected for an open hand pose from 15 different participants. The range of the orientation

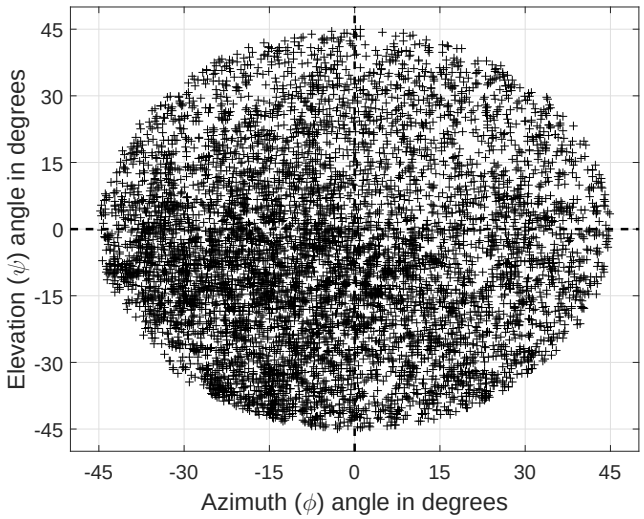


Fig. 5.9 Orientation space plot showing the orientation angles captured by *Dataset 2*.

angles captured by *Dataset 2* is restricted to a circular space with a radius of 45°, as shown in Fig. 5.9. This gives us an appropriate ratio for the number of samples against the variations within this defined orientation space. We use this dataset to show the experimental results that demonstrate the ability of our proposed ML-RF MtR method to apply marginalization through regression for estimating hand orientations. We note that, as can be seen in Fig. 5.9, the data has a bias due to prevalence of orientation movements that are easier to perform. In this work, we have not utilized this bias information, however we note in Section 5.3 that this bias can be useful for further improving the performance of the proposed methods. In our experimental validation, we found the average execution time of the proposed ML-RF MtR prediction framework to be 21.90 ms per frame in a Matlab implementation on 3.2 GHz Core-i5 CPU.

5.2.1 Comparison Methods

The proposed framework is compared with the method described in Chapter 4, which uses a single-layered single-variate Random Forest (SL-RF SV) with independence assumption on each hand orientation angle. We also compare with three different methods for marginalization of multi-layered Random Forest (ML-RF) regressors [21, 66]. These methods are referred to as ML-RF1, ML-RF2 and ML-RF3 herein, adapted from [21] and [66]. While the

methods proposed in [21] and [66] do not originally address the hand orientation regression problem, they provide methods for marginalizing a ML-RF in different domains. In our experimental validation, these three ML-RF comparison methods use a two-layered RF with a coarse latent variable classification in the first layer and expert orientation regression in the second layer. For a given CDF vector $\mathbf{d}_k$, the latent variable classification layer infers the posterior probability $p(r_n|\mathbf{d}_k)$ for each latent variable $r_n \in \{r_1, r_2, r_3, r_4, r_5\}$. These methods only differ in marginalization where ML-RF1 uses the predicted latent variable in the coarse layer to select the corresponding expert regressor for prediction, as defined by:

$$r_n^* = \arg\max_{r_n} p\left(r_n|\mathbf{d}_k\right), \tag{5.12}$$

$$(\phi^*, \psi^*) = \arg\max_{(\phi,\psi)} p\left(\phi, \psi|r_n^*, \mathbf{d}_k\right). \tag{5.13}$$

ML-RF2 uses posterior probabilities of each latent variable in the coarse layer as marginalization weights for predicted angles from each expert regressors, whereas ML-RF3 uses posterior probabilities from both the coarse and the expert layers to present the marginalized posterior probability. The mathematical formulation for predictions using ML-RF2 is given by:

$$(\phi^*, \psi^*) = \sum_{n=1}^{N} p\left(r_n|\mathbf{d}_k\right) \ \arg\max_{(\phi,\psi)} p(\phi, \psi|r_n, \mathbf{d}_k), \tag{5.14}$$

where $N = 5$ is the total number of expert regressors in the ML-RF model. Equations 5.15 and 5.16 show the formulation for making predictions using ML-RF3.

$$p(\phi, \psi|\mathbf{d}_k) = \sum_{n=1}^{N} p\left(r_n|\mathbf{d}_k\right) \ p(\phi, \psi|r_n, \mathbf{d}_k), \tag{5.15}$$

$$(\phi^*, \psi^*) = \arg\max_{(\phi,\psi)} p(\phi, \psi|\mathbf{d}_k). \tag{5.16}$$

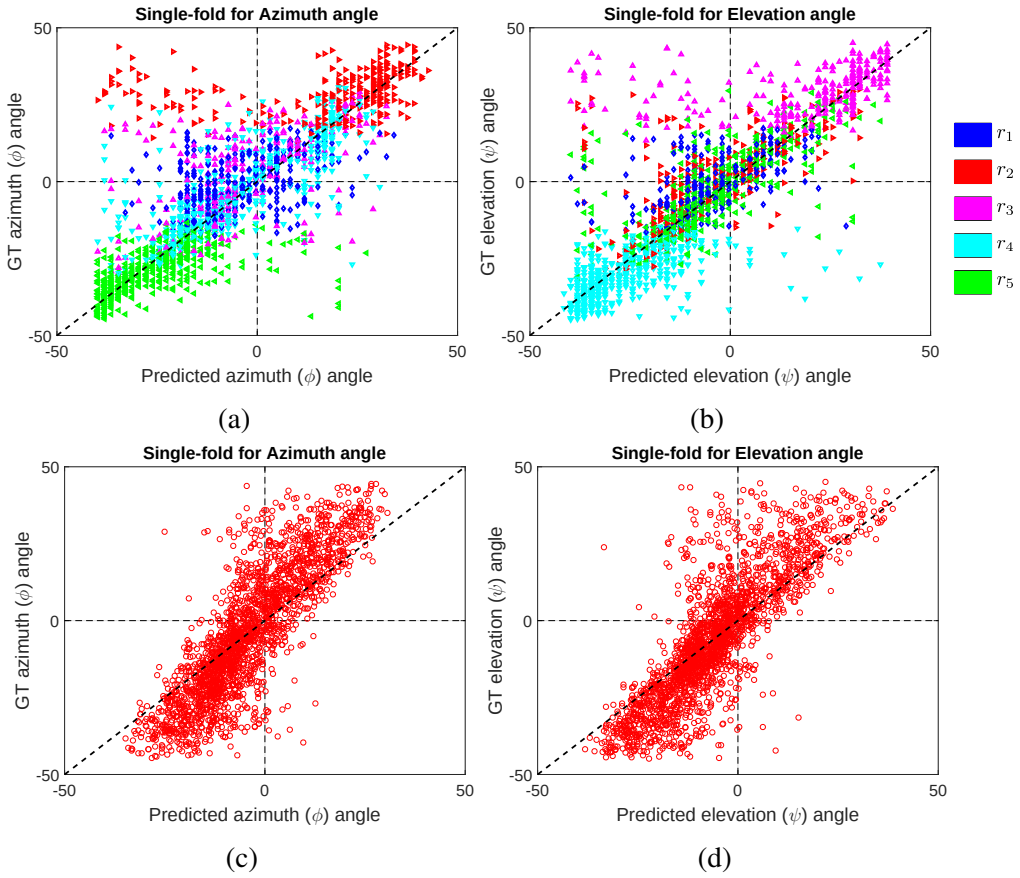


Fig. 5.10 Single-fold validation shows GT versus predicted angle results for azimuth and elevation angles using (a)-(b) ML-RF MtR and (c)-(d) SL-RF SV method from Chapter 4. It can be seen that the ML-RF MtR method infers hand orientation angles without a bias, which is the main source of error in SL-RF SV.

5.2.2 Parameter Selection

The proposed ML-RF MtR regression has a number of different training parameters. These include the number of trees (T), the depth of each tree (δ_t), the minimum number of samples in each leaf node (η_j), the number of features selected at each split node (ε) and the parameters α and β defining the latent variable-based label generation. In our experimental evaluation, we found that the parameters related to the RF classifier and regressors simultaneously improves the performance of all the comparison methods. Therefore, we empirically set these parameters to the following values for all experiments, $T = 100$, $\delta_t = 15$, $\eta_j = 5$ and $\varepsilon = 1$, $\alpha = 15°$ and $\beta = 45°$.

Evaluation method	**Method used**	**Azimuth (ϕ)**	**Elevation (ψ)**
Single-fold	ML-RF MtR	**8.12°**	**7.36°**
	SL-RF SV	9.43°	8.60°
	ML-RF1	8.80°	8.18°
	ML-RF2	11.31°	9.58°
	ML-RF3	8.69°	7.79°
User-specific	ML-RF MtR	**7.89°**	**7.29°**
	SL-RF SV	8.19°	7.94°
	ML-RF1	8.11°	7.45°
	ML-RF2	9.20°	8.50°
	ML-RF3	8.12°	7.72°

Table 5.1 MAE in degrees for experiments in Section 5.2.

5.2.3 Single-fold Validation

For this experiment, *Dataset 2* is randomly divided using 70% of the data for training and the remaining 30% for testing. Fig. 5.10 presents the predicted orientation angles using our proposed ML-RF MtR method and SL-RF SV from Chapter 4. These predicted angles are shown against their corresponding GT orientation angles, where in Fig.5.10 (a)-(b) we also show the corresponding latent variables using colors from Fig. 5.5. Furthermore, we also present the MAE of all comparison methods in Table 5.1. The proposed ML-RF MtR method outperforms the SL-RF SV in hand orientation inference due to its ability to learn expert regressors on subsets of a dataset. Furthermore, as opposed to training single-variate regressors for each orientation angles in Chapter 4, the proposed method utilizes multi-variate regressors to exploit the interdependence of the orientation angles. From Fig. 5.10 we observe that the ML-RF MtR method is able to infer orientation angles without introducing any bias, which is the main source of errors in SL-RF SV. Moreover, from Table 5.1, we note that the proposed ML-RF MtR method also outperforms the ML-RF1,

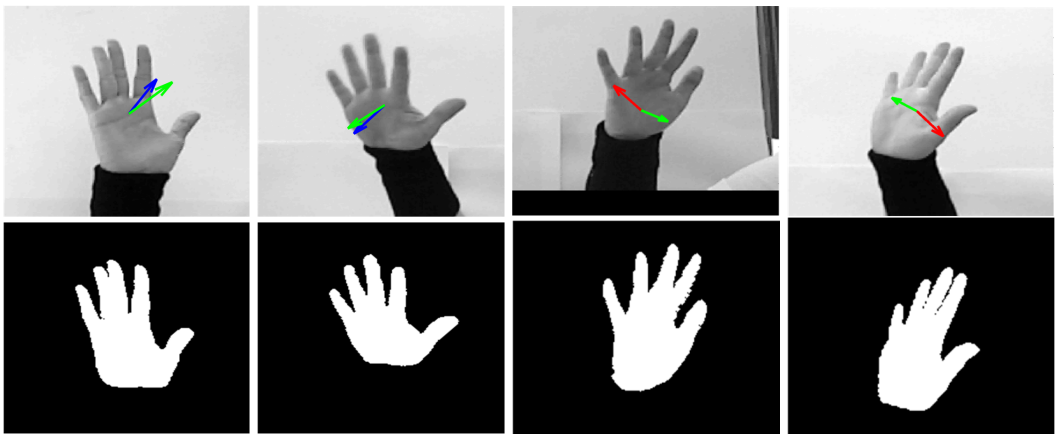


Fig. 5.11 Success and failure cases with normal vectors superimposed, where green shows GT normal vectors, blue shows predicted normal vector for success cases and red shows predicted normal vector for failure cases. It can be seen that the predicted normal vectors for failure cases are symmetrically opposite to the GT normal vectors. Silhouette images show how different orientations can result in similar silhouettes.

ML-RF2 and ML-RF3 in marginalization due to its ability to learn the marginalization weights with a regressor. Furthermore, as the marginalization weights are extracted using posterior probability distributions from expert regressors, therefore they also tend to address inaccuracies in these posterior probabilities. In contrast, previous methods directly use the posterior probabilities for marginalization which tend to underestimate the true posterior [119]. The errors in the proposed ML-RF MtR method come from symmetrically opposite latent variable spaces, i.e. hand facing left/right or up/down, as can be seen in Fig. 5.10 (a)-(b) at around $-40°$ and $40°$ GT orientation angles. This is due to the depth ambiguity of 2D silhouette images where two symmetric hand orientations produce similar results. Fig. 5.11 shows some example predictions with this symmetry problem. Nevertheless, these errors are few in number and do not affect the overall performance of our method as depicted in Table 5.1.

5.2.4 User-specific Validation

User-specific validation results of the proposed framework are shown in Table 5.1, where the training and testing is done using the same participant's data. This depicts an application scenario where a one-time model calibration will require the user to provide a user-specific

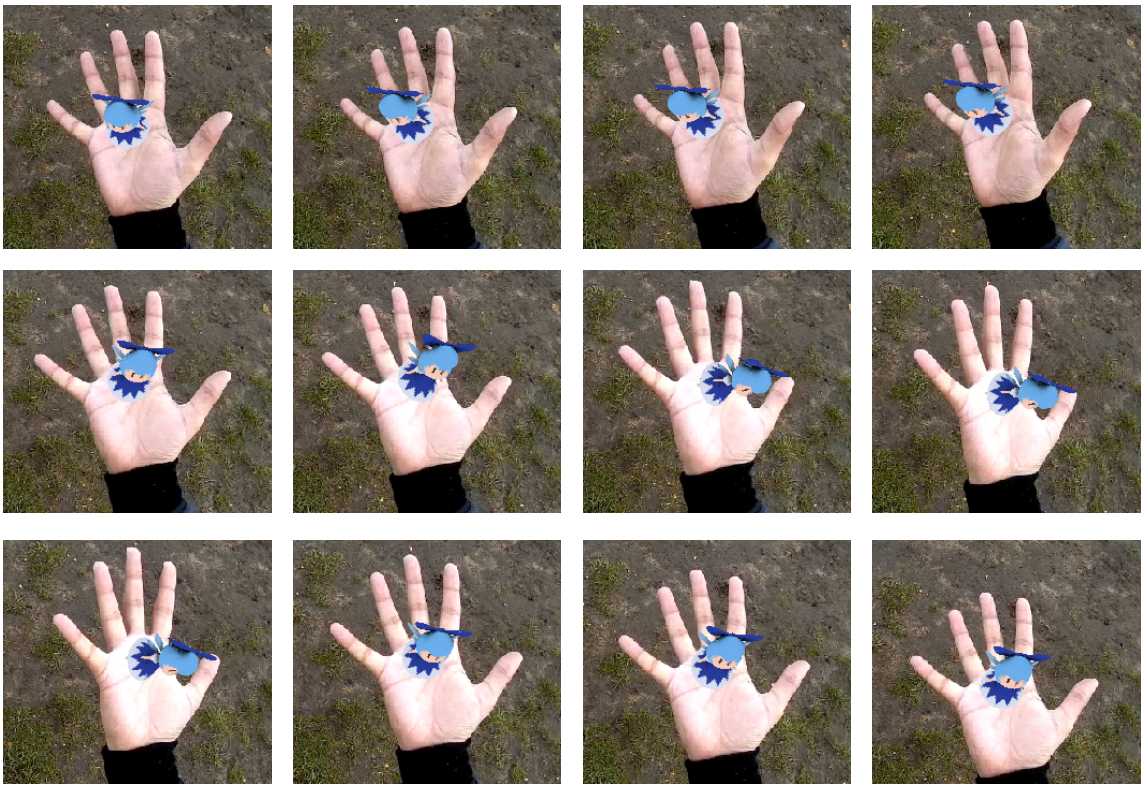


Fig. 5.12 Augmented Reality application in an egocentric setting. The hand orientation from our proposed method can be used for manipulating an augmented character on a hand palm. The sequence of hand orientation starts from the top left and ends at the bottom right image.

hand orientation dataset. Once trained, our proposed approach would be able to infer the hand orientation. For this validation, we divide each participants data into training (70%) and testing (30%) sets. From Table 5.1 we see that the proposed method performs even better than single-fold validation, as now the marginalization is fine-tuned for a particular user's hand where variations in shape and size are limited.

5.3 Application

The inferred hand orientation angles are useful in realizing a novel Augmented Reality application. We note that similar to Chapter 4, the hand orientation can provide manipulation-based interaction. In this work, we present an egocentric application scenario where a user wears a head mounted action camera. The video sequence from such scenario consists of a

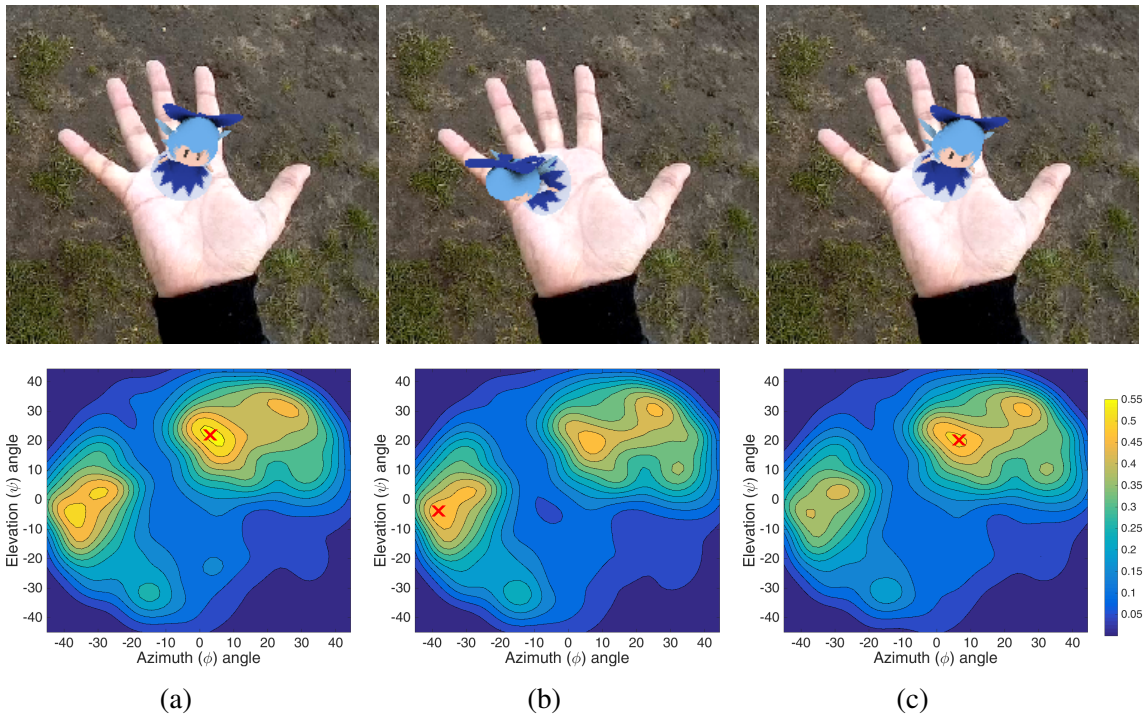


Fig. 5.13 Augmented Reality application sequence with a failure case. Top row shows the rendered augmented character on a continuous sequence of input images, whereas the bottom row shows the posterior probability $p(\phi, \psi|\mathbf{d}_k)$ from our proposed ML-RF MtR method. The MLE prediction is shown as a red x. It can be seen in (b) that even when our method fails, the posterior probability contains information for correct orientation angles.

number of variations that include radial lens distortion and abrupt hand and head movements. We demonstrate the application of our proposed method without calibrating the camera. The inferred hand orientation is used to provide interaction with an Augmented Reality character (as shown in Fig. 5.12), where the inferred orientations along with the location of the hand are used to render an augmented character on hand palm. A user can manipulate the character using different hand orientations inferred from the proposed ML-RF MtR method.

We also show the impact of errors from our proposed ML-RF MtR method on the Augmented Reality application (see Fig. 5.13). It can be seen that the posterior probabilities $p(\phi, \psi|\mathbf{d}_k)$ describe a symmetry problem, where multiple hypotheses for prediction exist. Even in the failure case in Fig. 5.13 (b), the posterior probability contains information regarding the correct orientation angles. As our method works independently on every image, therefore such errors are short-lived and are not accumulated over time. Furthermore, the

short-lived failures can be recovered by exploiting the temporal coherence in the posterior probabilities using a standard tracking technique [71, 120]. Additionally, the data has bias due to the presence of some orientation movements that are easier to do than others. This introduces a general prevalence in orientation movements. Our future work will aim to explore utilizing such prior knowledge regarding the orientation prevalence to improve the accuracy of the system.

5.4 Summary

This chapter proposed a novel marginalization method for multi-layered Random Forest regression of hand orientation. The proposed model was composed of two layers, where the first layer consisted of marginalization weights regressor while the second layer contained expert regressors trained on subsets of our hand orientation dataset. This method exploited the fact that learning from a number of variations in a dataset can be achieved by dividing the data into smaller subsets and using them to train expert models with better accuracy. To this end, we used a latent variable space to divide the hand orientation dataset into subsets. A Kullback-Leibler divergence-based optimization was used to estimate the weights that marginalized posterior probabilities from each expert regressors against a GT probability. Our main contribution came from the use of a marginalization weights regressor, which was trained on the marginalization weights. This regressor enabled our method to fine-tune the marginalization of the posterior probabilities during online prediction. Furthermore, we also proposed a method that is used for collecting hand orientation dataset, namely, *Dataset 2*, *3* and *4*. This method provided feedback to participants, which guided them to completely cover the possible hand orientation variations. Our proposed method outperformed the method previously described in Chapter 4 and state-of-the-art for both hand orientation inference and multi-layered Random Forest marginalization with an average error of 7.74° for single-fold validation and 7.59° for a user-specific scenario. We also showed an Augmented Reality application in an egocentric scenario. One limitation of the proposed multi-layered Random Forest regression is that it required an explicit definition of subsets with clear boundaries.

This limited the number of expert regressors as defining more subsets results in fewer samples in each subset, which produces under fitting models. Moreover, in some domains the optimal latent variable space is unknown, hence limiting the use of the proposed ML-RF MtR method. We address these challenges in the next chapter, where we propose a staged probabilistic regression method that is capable of defining latent variable space as it learns.

Chapter 6

SPORE: Staged Probabilistic Regression for Hand Orientation and Pose Estimation

This chapter proposes the staged probabilistic regression (SPORE) method which consists of multiple expert regressors, each one of them learning a subset of variations from the dataset [31]. Inspired by Boosting, the novelty of our method comes from the staged probabilistic learning, where each stage consists of training and adding an expert regressor to the intermediate ensemble of expert regressors. SPORE addresses the limitation of explicit definition of latent variable space faced by the proposed ML-RF MtR method in Chapter 5. The proposed SPORE method utilizes a staged learning approach where after training each stage, the evaluation of the intermediate model is used to extract the latent variable space for the next stage. Unlike Boosting, we utilize the marginalization weights regressor proposed in Chapter 5, to marginalize the posterior prediction probabilities from each expert regressor. We extend and evaluate our proposed framework for simultaneously inferring hand orientation and pose. Our proposed method proves to be more accurate in comparison to the state-of-the-art of both hand orientation inference and multi-layered Random Forest (ML-RF) marginalization. We also show application scenarios where the hand orientation and pose can be combined to enable interaction with an augmented character and a computer game.

Moreover, experimental results reveal that simultaneously learning hand orientation and pose from 2D monocular images significantly improves the performance of pose classification.

6.1 Staged Probabilistic Regression

In our proposed method, we utilize a ML-RF composed of two layers, where the first layer consists of a single marginalization weights regressor and the second layer is composed of an ensemble of expert regressors trained on subsets of the hand orientation dataset. We introduce a staged learning method that trains and incrementally adds the expert regressors to the model. The flowchart of the training and prediction framework for SPORE is presented in Fig. 6.1. Algorithm 1 and 2 describe the pseudocode for the training and prediction algorithms for SPORE, which appear in Section 6.1.1. In the proposed framework, each added expert regressor is trained on the samples that the existing expert regressors have difficulty in learning. We achieve this by combining the existing models using marginalization weights and evaluating the accuracy of the model after each training stage. Based on a threshold error, we identify the harder regression problems after each stage and use these samples to train the next expert regressor. The chosen approach enables us to use our regression-based marginalization framework without defining subsets using latent variable boundaries as defined in Chapter 5. When all expert regressors have been trained, the posterior probabilities corresponding to each sample in the training set are acquired from each of the trained expert regressors. We apply a Kullback-Leibler divergence-based optimization technique, proposed in Chapter 5, that estimates the marginalization weights for estimating marginal probability distribution from the given ensemble of expert regressors. We use these marginalization weights to train a marginalization weights regressor which enables us to combine the ensemble of expert regressors. This staged learning approach allows us to achieve higher accuracy as compared to previously proposed marginalization methods as well as a single regressor-based approach. We now describe the SPORE approach in detail.

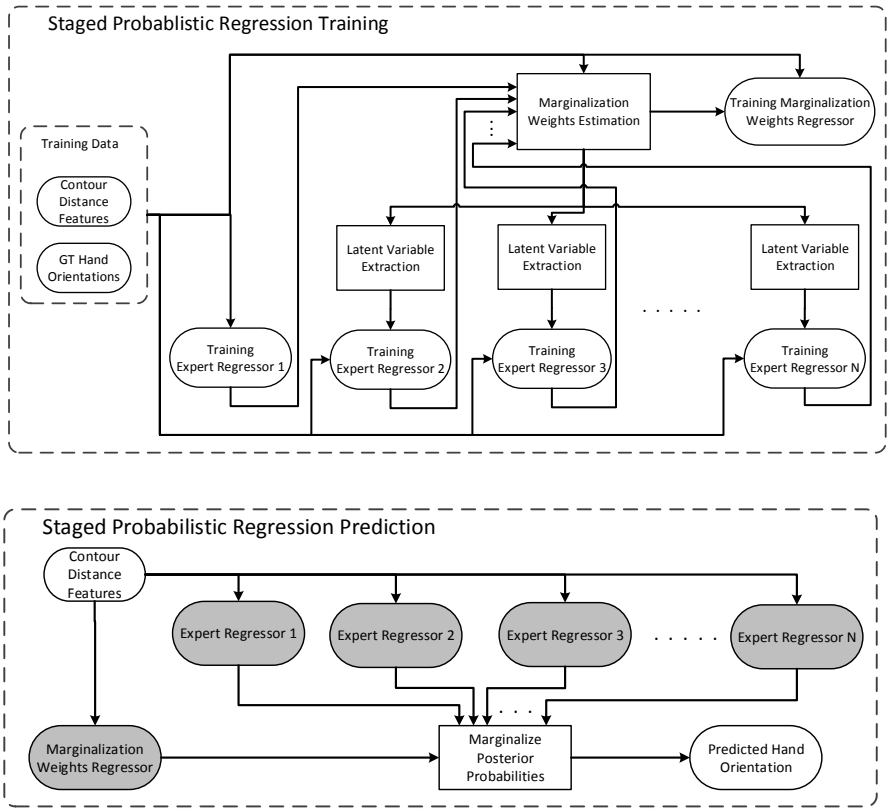


Fig. 6.1 Flowchart shows the SPORE training and prediction framework. The trained regressors are highlighted with a solid gray background.

6.1.1 Marginalization of Multiple Expert Regressors

In our proposed method, the ensemble of expert regressors consists of a set of multi-variate Random Forest (RF) regressors that are trained on subsets of our hand orientation dataset. This ensemble of expert regressors enables better generalization in the presence of a number of variations in the dataset. The subsets of the dataset are defined based on latent variable representations that are generated using the intermediate model evaluations. Given an input Contour Distance Feature (CDF) vector $\mathbf{d}_k$, each expert regressor infers the posterior probability $p(\phi, \psi | r_n, \mathbf{d}_k)$ for a given latent variable r_n.

The proposed expert regression layer contains an ensemble of trained expert regressors, where the task of marginalization is to estimate their combined marginal probability $p(\phi, \psi | \mathbf{d}_k)$ that is used to infer orientation angles $\mathbf{o} = (\phi, \psi)$ for a given input feature vector

Algorithm 1: Training algorithm for SPORE.

Input: $\mathcal{U}_{all} = \{(\mathbf{d}_1, \mathbf{o}_1), \cdots, (\mathbf{d}_k, \mathbf{o}_k), \cdots, (\mathbf{d}_K, \mathbf{o}_K)\}$, N, σ
% N is the number of stages
% σ is the error threshold

Output: ($\mathbf{ER}_n$, **MR**)
% $\mathbf{ER}_n$ are N Expert Regressors
% **MR** is the Marginalization Weights Regressor

$n \leftarrow 1$ % Starting stage
$\{r_n(k)\}_{k=1}^{K} \leftarrow 1$ % Latent variable selecting all samples
$\mathcal{U}_{sel} \leftarrow$ selectSubset($\mathcal{U}_{all}$, r_n) % Select initial subset of $\mathcal{U}_{all}$

% Training $\mathbf{ER}_n$
for $n \leftarrow 1$ **to** N **do**
  $\mathbf{ER}_n \leftarrow$ Train($\mathcal{U}_{sel}$) % Train stage n using selected subset
  if $n = 1$ **then**
    $p(\mathbf{o}_k|r_n, \mathbf{d}_k) \leftarrow$ Predict($\mathbf{d}_k$, $\mathbf{ER}_n$) % Get posterior probabilities
    $\mathbf{o}_p(k) \leftarrow \arg\max_{\mathbf{o}_k} p(\mathbf{o}_k|r_n, \mathbf{d}_k)$
  else
    for $m \leftarrow 1$ **to** n **do**
      $p(\mathbf{o}_k|r_m, \mathbf{d}_k) \leftarrow$ Predict($\mathbf{d}_k$, $\mathbf{ER}_m$)
    end
    $\omega_{nk} \leftarrow$ getMarginalizationWeights($p(\mathbf{o}_k|r_n, \mathbf{d}_k)$) % Described in Section 6.1.3
    $p(\mathbf{o}_k|\mathbf{d}_k) \leftarrow \sum_{m=1}^{n} p(\mathbf{o}_k|r_m, \mathbf{d}_k)\,\omega_{mk}$ % Marginalize probabilities described in Section 6.1.1
    $\mathbf{o}_p(k) \leftarrow \arg\max_{\mathbf{o}_k} p(\mathbf{o}_k|\mathbf{d}_k)$
  end
  % Define latent variable for next stage described in Section 6.1.2
  if $|\mathbf{o}_p(k) - \mathbf{o}_k| > \sigma$ **then**
    $r_n(k) \leftarrow 1$
  else
    $r_n(k) \leftarrow 0$
  end
  $\mathcal{U}_{sel} \leftarrow$ selectSubset($\mathcal{U}_{all}, r_n$)
end

% Training **MR**
for $n \leftarrow 1$ **to** N **do**
  $p(\mathbf{o}_k|r_n, \mathbf{d}_k) \leftarrow$ Predict($\mathbf{d}_k$, $\mathbf{ER}_n$) % Get posterior probabilities
  $\omega_{nk} \leftarrow$ getMarginalizationWeights($p(\mathbf{o}_k|r_n, \mathbf{d}_k)$)
  $\mathcal{W}_{all} \leftarrow \{(\mathbf{d}_1, \omega_{n1}), \cdots (\mathbf{d}_K, \omega_{nK})\}$ % Define training set for **MR**
  MR $\leftarrow$ Train($\mathcal{W}_{all}$)
end
return $\mathbf{ER}_n$, **MR**

$\mathbf{d}_k$. This marginal probability is defined as:

$$p(\phi, \psi|\mathbf{d}_k) = \sum_{n=1}^{N} p(\phi, \psi|r_n, \mathbf{d}_k)\,\omega_{nk}, \tag{6.1}$$

where ω_{nk} are weights corresponding to each latent variable such that $\sum_{n=1}^{N} \omega_{nk} = 1$ and N is the total number of expert regressors. This marginal probability can be used for maximum

Algorithm 2: Prediction algorithm for SPORE.

Input: $\mathbf{d}$, $\mathbf{ER}_n$, $\mathbf{MR}$, N
% $\mathbf{d}$ is the input Contour Distance Feature vector
% $\mathbf{ER}_n$ are N Expert Regressors
% $\mathbf{MR}$ is the Marginalization Weights Regressor

Output: $\mathbf{o}$
% $\mathbf{o} = (\phi, \psi)$ is a vector of predicted orientation angles

$\mathbf{o} \leftarrow \varnothing$
$\omega_n \leftarrow$ Predict($\mathbf{d}$, $\mathbf{MR}$) % Predict Marginalization Weights
for $n \leftarrow 1$ **to** N **do**
  $p(\mathbf{o}|r_n, \mathbf{d}) \leftarrow$ Predict($\mathbf{d}$, $\mathbf{ER}_n$) % Get posterior probabilities
end
$p(\mathbf{o}|\mathbf{d}) \leftarrow \sum_{n=1}^{N} p(\mathbf{o}|r_n, \mathbf{d})\, \omega_n$ % Marginalize posterior probabilities
$\mathbf{o} \leftarrow \arg\max_{\mathbf{o}} p(\mathbf{o}|\mathbf{d})$
return $\mathbf{o}$

likelihood estimate (MLE) as:

$$(\phi^*, \psi^*) = \underset{(\phi,\psi)}{\arg\max}\, p(\phi, \psi|\mathbf{d}_k)\,. \tag{6.2}$$

6.1.2 Latent Variable Generation using Intermediate Models

In our proposed work for SPORE, we do not explicitly define the latent variable space, as we defined in Chapter 5. However, herein we rely on intermediate model evaluations for extracting a latent variable r_n, which defines the subset used for training the expert regressor in the n^{th} stage. We start training the first expert regressor using all samples in the dataset. Following this, we train and add additional expert regressors to the ensemble using subsets of the dataset defined by the corresponding latent variable r_n. For each training sample in the dataset, we determine if it belongs to the latent variable r_n as:

$$r_n(k) = \begin{cases} 1 & \text{if } |\mathbf{o}_p(k) - \mathbf{o}_k| > \sigma, \\ 0 & \text{otherwise,} \end{cases} \tag{6.3}$$

where $\mathbf{o}_p(k)$ are the orientation angles which are predicted by marginalizing intermediate model probabilities using Equation 6.1 and $\mathbf{o}_k$ are the ground truth (GT) orientation angles. σ is an adjustable threshold and $r_n(k) \in \{0, 1\}$ determines if the given sample belongs to the latent variable r_n for the n^{th} stage.

This method has two advantages over the previously proposed latent variable-based training described in Chapter 5. First, the proposed method relies on the model to define and use subsets, which might be useful in cases where optimal latent variable-based subset definitions are difficult or not well defined a *priori*. Second, our proposed incremental learning method is capable of defining latent variables with overlapping boundaries ensuring complete training of expert regressors, in cases where datasets are small and dividing them into subsets can result in shallow under fitting models.

In the subsequent sections, we present an overview of the method for estimating the GT marginalization weights ω_{nk} from trained expert models. Our approach is similar to the marginalization weights estimation presented in Chapter 5. However SPORE can work with up to N expert regressors. In our experimental validation we determine the number of stages N by empirical parameter optimization experiments described in Section 6.2.2.

6.1.3 Marginalization through Regression

We generalize the marginalization through regression method, proposed in Chapter 5, for N expert regressors. Given the GT probability $p(\phi_k, \psi_k)$ and the corresponding posterior probabilities $p(\phi, \psi | r_n, \mathbf{d})$, we utilize the optimization method from Chapter 5 in which the marginalization error is based on Kullback-Leibler divergence [118]. The error is optimized to estimate the GT marginalization weights ω_{nk} for all latent variables $r_n \in \{r_1, r_2, r_3 \cdots r_N\}$. This error is defined as:

$$E = \iint p(\phi_k, \psi_k) \log \frac{p(\phi_k, \psi_k)}{p(\phi, \psi | \mathbf{d}_k)} d\phi d\psi. \tag{6.4}$$

The partial derivative of E w.r.t ω_{nk} is defined as:

$$\frac{\partial E}{\partial \omega_{nk}} = - \iint \frac{p(\phi_{gt}, \psi_{gt})\, p(\phi, \psi | r_n, \mathbf{d})}{\sum_{n=1}^{N} p(\phi, \psi | r_n, \mathbf{d})\, \omega_{nk}} d\phi d\psi. \tag{6.5}$$

We estimate the optimal weights ω_{nk} by utilizing the partial derivatives in gradient descent as:

$$\nabla E = \left[\frac{\partial E}{\partial \omega_{1k}}, \frac{\partial E}{\partial \omega_{2k}}, \frac{\partial E}{\partial \omega_{3k}} \cdots \frac{\partial E}{\partial \omega_{Nk}} \right], \tag{6.6}$$

for which the optimization is iteratively evolved for a solution given by:

$$\omega_{nk}^{\upsilon+1} = \omega_{nk}^{\upsilon} - \lambda \nabla E^{\upsilon}, \tag{6.7}$$

where λ is the step size along the negative gradient direction and υ is the iteration number.

At this stage, we have the optimal weights fit to the GT probabilities defined by GT orientation angles. These weights are required to train the marginalization weights regressor that produces the weights ω_n during online prediction.

Marginalization Weights Regressor

We use a multi-variate RF regressor, proposed in Chapter 5, to learn the mapping of CDF $\mathbf{d}_k$ onto marginalization weights ω_{nk}. This regressor is used during prediction to infer marginalization weights ω_{nk} for marginalizing the posterior probabilities $p(\phi, \psi | r_n, \mathbf{d}_k)$ from each expert regressors using Equation 6.1.

6.1.4 Extension to Estimate Orientation and Pose

The proposed SPORE method can be extended to simultaneously infer the hand orientation and pose angles. To achieve this, we utilize a hand orientation and pose dataset which contains the CDF ($\mathbf{d}_k$), the corresponding categorical hand pose label (χ_k) and the orientation angles ($\mathbf{o}_k$). We introduce the pose classification into each expert regressor by including the discrete posterior probability distributions $p(\chi | \mathbf{d}_k)$ in the leaf nodes. Training of the extended model is driven by both pose classification as well as orientation regression data. We achieve this by using an information gain Q_s, which is determined by:

$$Q_s = (1-\beta) Q_r + \beta Q_c, \tag{6.8}$$

where Q_r is the orientation regression information gain, Q_c is the pose classification information gain and $\beta \in \{0,1\}$ is a random variable selected with probability $p(\beta)$. We use standard classification and regression information gain as defined in Chapter 3.

Given the additional pose classification task, we define the latent variable space r_n by modifying Equation 6.3 with an additional term as:

$$r_n(k) = \begin{cases} 1 & \text{if } |\mathbf{o}_p(k) - \mathbf{o}_k| > \sigma \text{ or } \chi_p(k) \neq \chi_k, \\ 0 & \text{otherwise,} \end{cases} \tag{6.9}$$

where $\chi_p(k)$ and χ_k are the predicted and GT hand poses respectively. The additional criteria related to hand poses in Equation 6.9 identifies samples for which the existing intermediate model has difficulty in inferring the hand pose.

For an input CDF vector $\mathbf{d}_k$, each expert model now additionally infers the posterior probability $p(\chi|r_n,\mathbf{d}_k)$. We marginalize these posterior probabilities using:

$$p(\chi|\mathbf{d}_k) = \sum_n p(\chi|r_n,\mathbf{d}_k)\,\rho_{nk}, \tag{6.10}$$

where ρ_{nk} are weights corresponding to each latent variable for the classification posterior probabilities $p(\chi|r_n,\mathbf{d}_k)$ and $\sum_n^N \rho_{nk} = 1$. We estimate these marginalization weights using discrete version of energy E defined as:

$$E_c = \sum_{\chi} p(\chi_k) \log \frac{p(\chi_k)}{p(\chi|\mathbf{d}_k)}. \tag{6.11}$$

The partial derivatives w.r.t ρ_{nk} can be defined using E_c as:

$$\frac{\partial E_c}{\partial \rho_{nk}} = -\sum_{\chi} \frac{p(\chi_k)\,p(\chi|r_n,\mathbf{d}_k)}{\sum_{n=1}^N p(\chi|r_n,\mathbf{d}_k)\,\rho_{nk}}. \tag{6.12}$$

We use gradient descent to estimate the optimal weights ρ_{nk} for the classification posterior probabilities. We concatenate the marginalization weights for pose classification ρ_{nk} and

orientation regression ω_{nk} together to train a marginalization weights regressor that infers both weights simultaneously.

6.2 Experimental Validation

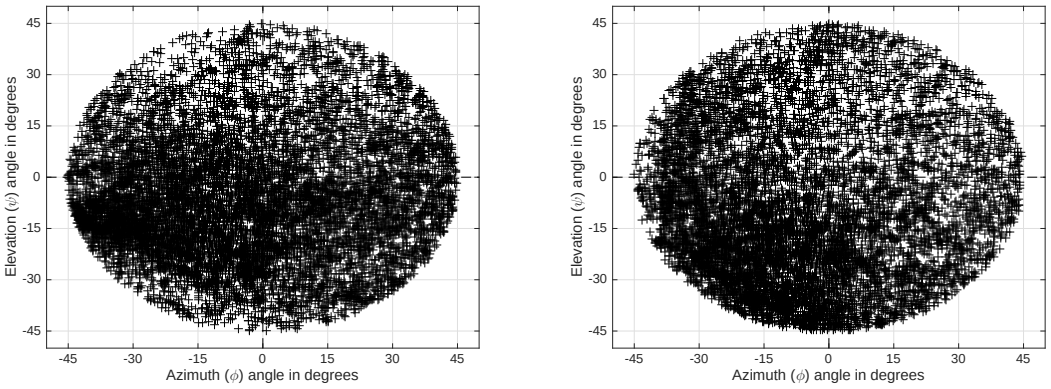
We evaluate our proposed SPORE method using two datasets collected from 22 participants. The first dataset, referred to as *Dataset 3* herein, contains 9414 samples captured for an open hand pose from 22 different participants. The second dataset, referred to as *Dataset 4* herein, contains 8675 samples captured using four different hand poses (shown in Fig. 6.2) from 10 different participants. The different hand poses used for experimental validation are limited, however, they demonstrate the applicability of the proposed method in scenarios where multiple hand poses are required. The range of the orientation angles captured by these datasets is restricted to a circular space defined by $\sqrt{\phi^2 + \psi^2} \leq 45°$, as shown in orientation space plots in Fig. 6.3. This gives us an appropriate ratio for the number of samples against the variations within the defined orientation space. The four hand poses used in this work are selected as they follow our planar hand assumption, which enables us to extract reliable GT hand orientation using the method described in Chapter 5. Moreover, these hand poses present different shapes of the hand, which can be used to trigger interaction with Augmented Reality objects. The χ_3 and χ_4 hand poses are selected to show the importance of hand orientation for estimating hand pose. These two postures, under certain viewpoints, exhibit similar hand shape especially when the fifth digit or pinky finger is partially occluded (see Fig. 6.12 (c)). As shown in Table 6.3, the SPORE method trained using both orientation and pose data is able to infer these two postures accurately. The proposed method can be easily generalized to use additional hand poses, where additional poses can be added along with their labels, e.g. χ_5. The prediction framework of the proposed SPORE method has an average execution time of 30.80 ms per frame for orientation estimation and 37.70 ms for simultaneous orientation and pose estimation. These execution times are acquired using a Matlab implementation on 3.2 GHz Core-i5 CPU.

(a) χ_1 (b) χ_2 (c) χ_3 (d) χ_4

Fig. 6.2 Four hand postures, along with their corresponding labels, used for multiple pose experimental validation. (a) shows an open hand pose used for single pose experimental validation of SPORE.

6.2.1 Comparison Methods

The proposed method is compared with a previous method for hand orientation regression that uses a single-layered single-variate Random Forest (SL-RF SV) with independence assumption on each hand orientation angle. We also compare with five different methods for marginalization of ML-RF regressors [21, 30, 66]. Furthermore, as SPORE is inspired from Adaboost, we compare the proposed method with Random Forest with Adaboost [121]. Our work in Chapter 5, referred to as ML-RF MtR herein, is closely related to SPORE. This method also utilized a multi-layered Random Forest, where the first layer consisted of a single marginalization weights regressor and the second layer contained five expert regressors. The expert regressors in ML-RF MtR were trained on subsets of the orientation dataset defined using a simple observation that the hand can be oriented (i) fronto-parallel, or facing (ii) right, (iii) left, (iv) upwards or (v) downwards with respect to the camera. Marginalization weights for the expert regressors were extracted using posterior probabilities and a Kullback-Leibler divergence-based optimization as described in Section 5.1.3. ML-RF MtR differs from our proposed SPORE method in terms of the explicit definition of the five latent variables for defining subsets of the training data. In contrast, SPORE relies on the learned models to define the next most suitable latent variable space which has a number of advantages, as discussed in Section 6.2.3. Moreover, we also utilize the k-means clustering algorithm with five clusters to define latent variables for training our proposed ML-RF MtR method. This method, referred to as ML-RF MtR K-means herein, enables us to understand



(a) (b)

Fig. 6.3 Orientation space plots showing the orientation angles captured by (a) *Dataset 3* and (b) *Dataset 4*.

whether SPORE is learning meaningful latent variable-based representations or if it is just doing clustering at each stage. The other ML-RF marginalization methods are referred to as ML-RF1, ML-RF2 and ML-RF3 herein which are adapted from [21] and [66]. These methods also rely on the same explicit definition of latent variables as in ML-RF MtR. As defined in Section 5.2, all three ML-RF comparison methods use a two-layered RF with a coarse latent variable classification in the first layer and expert orientation regression in the second layer. These methods only differ in marginalization where ML-RF1 uses the predicted latent variable in the coarse layer to select the corresponding expert regressor for prediction. ML-RF2 uses posterior probabilities of each latent variable in the coarse layer as marginalization weights for predicted angles from each expert regressor, whereas ML-RF3 uses posterior probabilities from both the coarse and the expert layers to present the marginalized posterior probability.

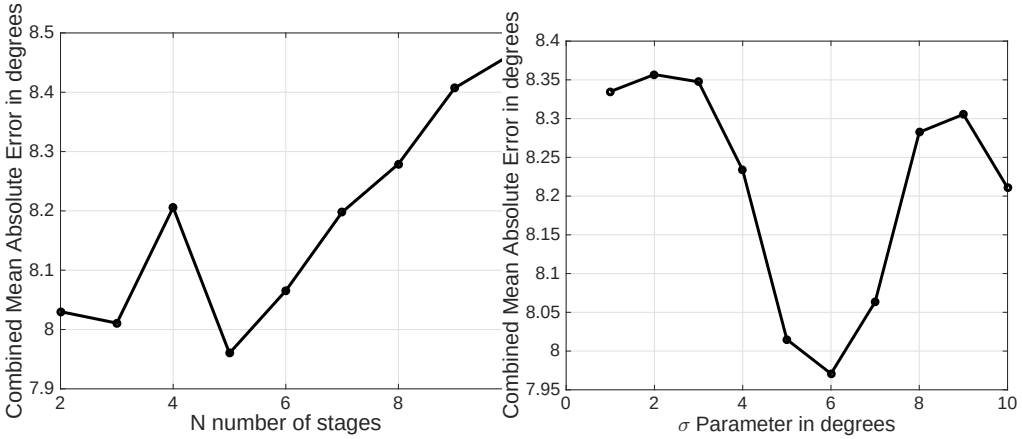
We evaluate the extension of our proposed method by simultaneously estimating the orientation and pose using the *Dataset 4*. To show the role of hand orientation in improving the pose classification performance, we compare this extension with a Random Forest Classifier (RF Clf) that infers hand pose only. For this extension, we also compare the orientation inference with all the comparison methods mentioned above. We further discuss the results of these comparisons in Section 6.2.4.

6.2.2 Parameter Optimization

The proposed SPORE method has different training parameters which include, (i) the number of trees (T), (ii) the depth of each tree (δ_t), (iii) the minimum number of samples in each leaf node (η_j), (iv) the number of features selected at each split node (ε), (v) the number of stages (N), (vi) the latent variable generation parameter σ and (vii) the probability $p(\beta)$ for selecting information gain for the extension of the proposed method for simultaneous hand orientation and pose inference. All comparison methods utilize RF, therefore we empirically set the values of the related parameters as, $T = 100$, $\delta_t = 10$, $\eta_j = 20$, $\varepsilon = 1$. As the proposed SPORE method is not dependent on the number of predefined subsets, therefore any number of stages N can be used. To evaluate the optimal values for N and σ, we use a single-fold validation with *Dataset 3*, 70% of which is used for training and 30% for testing. The optimal values for $p(\beta)$ are determined using a single-fold validation with *Dataset 4*.

The Combined Mean Absolute Error (CMAE) with varying number of stages N is shown in Fig. 6.4 (a). It can be seen that SPORE with $N = 5$ stages presents the minimum CMAE for both azimuth (ϕ) and elevation (ψ) angles. The error increases for $N > 5$, which means that the variations contained in our hand orientation dataset can be optimally captured by $N = 5$. We choose $N = 5$ for the rest of the experimental validation. Fig. 6.4 (b) shows the CMAE with varying σ threshold in Equation 6.3. σ acts as a threshold for defining the subset of training data for the next stage. We observe that if σ is too low, i.e. $\sigma \approx 0$, then the subsequent stages will all be trained using all training samples, thus not targeting to learn from specific variations. On the contrary, if σ is set too high, i.e. $\sigma > 10°$, then the latent variable space will not be fully defined for subsequent stages, hence resulting in under fitting models. We observe that $\sigma = 6°$ maintains a good balance for selecting harder samples for training subsequent stages while also yielding the least CMAE in Fig. 6.4 (b). Therefore, we select this value for the rest of the experimental validation.

The extension of our proposed SPORE method to classify hand pose additionally depends on probability $p(\beta)$ for selecting classification or regression information gain for training. We present the effect of varying this probability on hand orientation and pose inference in Fig. 6.5, which shows that selecting regression information gain more often than classification



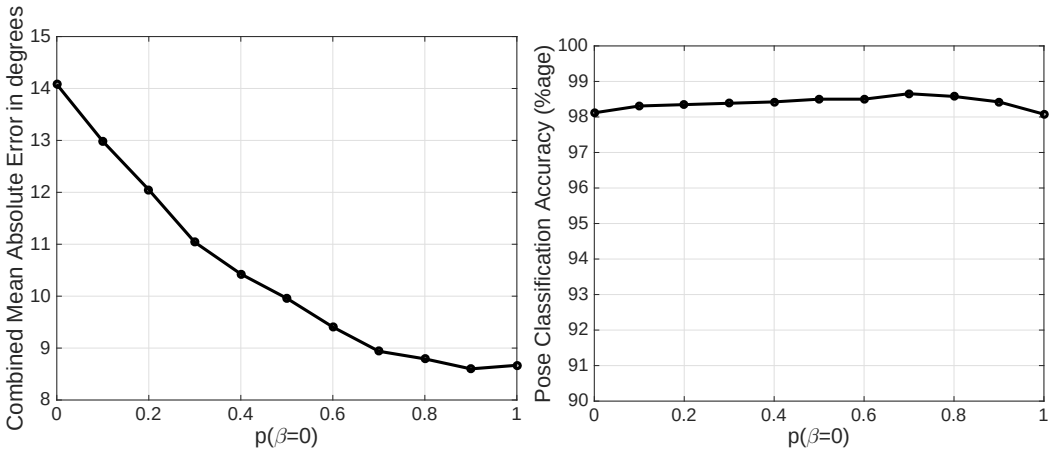
(a) (b)

Fig. 6.4 Parameter optimization using *Dataset 3*, shows evaluation of the proposed SPORE method for hand orientation inference. The figures present CMAE for (a) varying number of stages N and (b) varying σ threshold in Equation 6.3.

information gain (i.e. $p(\beta = 0) > 0.5$) yields better performance for both hand orientation and pose inference. It can also be seen that the pose classification is solved even when no classification information gain is employed (i.e. $p(\beta = 0) = 1$). The reason for this is, the information for each pose is well encoded within the CDF and hand orientation. In our experimental validation we use $p(\beta = 0) = 0.8$. It means that at each split node, regression information gain is selected more frequently than classification information gain. As we will further demonstrate in Section 6.2.4, the hand orientation information can significantly improve pose classification results because with orientation the SPORE model is able to build a better understanding of the hand pose dataset. This is specifically seen for poses χ_3 and χ_4 in Tables 6.3 and 6.4.

6.2.3 Experimental Validation using Single Pose Dataset

Our proposed hand orientation inference method is evaluated using the *Dataset 3*. We perform single-fold validation by randomly dividing 70% of the data into training set and using the remaining 30% for testing. Table 6.1 shows the Mean Absolute Error (MAE) in degrees for the single-fold evaluation using SPORE method and the comparison methods.



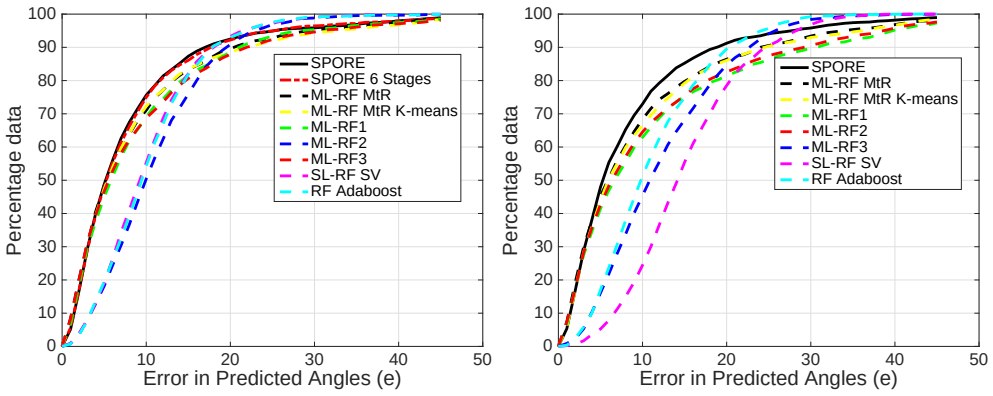
(a) (b)

Fig. 6.5 Parameter optimization for $p(\beta = 0)$ using *Dataset 4*, shows evaluation of the proposed SPORE method with hand orientation and pose estimation extension. (a) presents CMAE for orientation inference and (b) shows the accuracy of pose classification against varying probability $p(\beta = 0)$ of selecting regression information gain.

Method used	**Azimuth (ϕ)**		**Elevation (ψ)**	
		p-value		p-value
SPORE	**8.42°**	-	**7.38°**	-
ML-RF MtR	9.65°	**0.00**	7.81°	**0.13x10^{-10}**
ML-RF MtR K-means	9.80°	**0.04x10^{-10}**	8.14°	**0.14x10^{-8}**
SL-RF SV	11.58°	**0.25x10^{-8}**	8.75°	**0.00**
RF Adaboost [121]	11.54°	**0.72x10^{-10}**	9.06°	**0.00**
ML-RF1	10.24°	**0.22x10^{-5}**	8.02°	**0.00**
ML-RF2	12.82°	**0.20x10^{-3}**	9.12°	**0.11x10^{-2}**
ML-RF3	10.45°	**0.10x10^{-20}**	8.13°	**0.15x10^{-18}**
SPORE (*Dataset 1*)	15.46°	-	13.37°	-

Table 6.1 MAE in degrees for single pose experimental validation in Section 6.2.3.

Furthermore, the percentage of data that lies under a given error (e) is shown in Fig. 6.6 (a). The percentage data in this plot is estimated using Equation 3.21.



(a) (b)

Fig. 6.6 Percentage data versus error in prediction shows the percentage of data that lies below a given error in prediction for the single-fold validation using (a) *Dataset 3* and (b) *Dataset 4*.

Our results demonstrate that SPORE outperforms the other existing methods in ML-RF marginalization as well as hand orientation inference. The proposed method also outperforms the Random Forest with Adaboost which lacks a probabilistic formulation, resulting in higher MAE. In contrast, the proposed method is formulated using probabilities, where the complex mapping between each stage and the input features is learned. We further observe from Fig. 6.6 (a) that the proposed SPORE method performs better with 78% of data lying in under 10° of error. Our analysis shows that at around 20° of error, the RF Adaboost, ML-RF2 and SL-RF SV contain higher percentage data than any other method. This is due to the fact that all other comparison methods, including the proposed SPORE, are affected by the symmetry problem for around 10% of the data. The symmetry problem arises as a result of depth ambiguity in 2D monocular images, where multiple hand orientations can produce the same contour. It affects the regressors where for a given hand contour, the regressors infer symmetrically opposite hand orientations. RF Adaboost, ML-RF2 and SL-RF SV infer only a few symmetrically opposite hand orientations. As these methods rely on the weighted sum of regressor predictions or a prediction from a single regressor, therefore the variations due to the symmetry problem result in introducing a model bias. This results in greater MAE for these methods in Table 6.1. In Fig. 6.6 (a), we also show the performance of the SPORE model with $N = 6$ stages. It shows a slight decrease in performance, indicating that

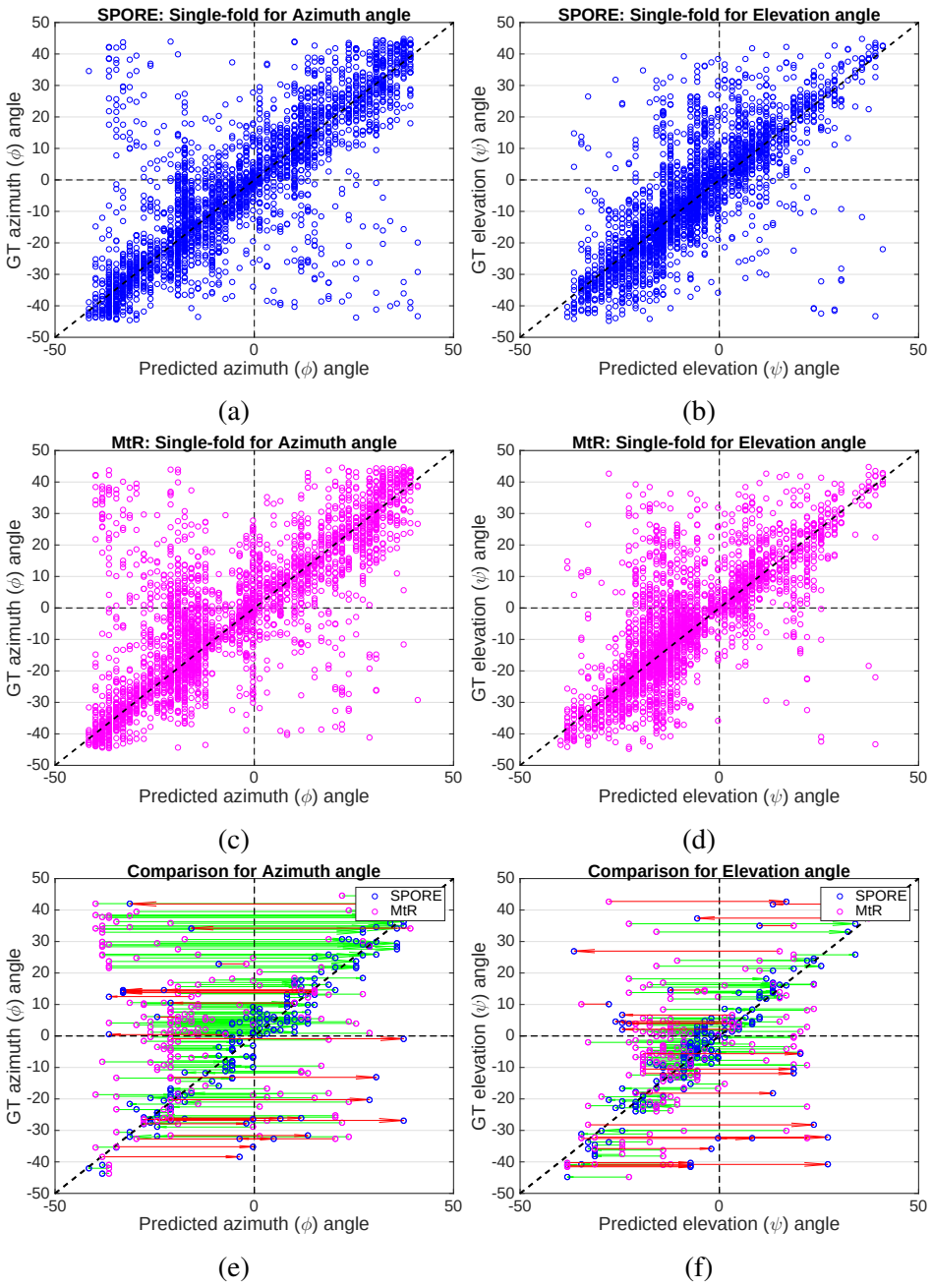


Fig. 6.7 GT versus predicted orientation angle plots showing results for (a)-(b) the proposed SPORE method and (c)-(d) the ML-RF MtR method proposed in Chapter 5. (e)-(f) shows the errors in ML-RF MtR that were corrected by SPORE (green arrows) and the correct predictions by ML-RF MtR that were incorrectly inferred by SPORE (red arrows). In (e)-(f), we label a sample as being correctly (incorrectly) predicted if the absolute error is less (greater) than 15°. The larger number of green lines compared to red shows that SPORE improves estimation for the majority of samples.

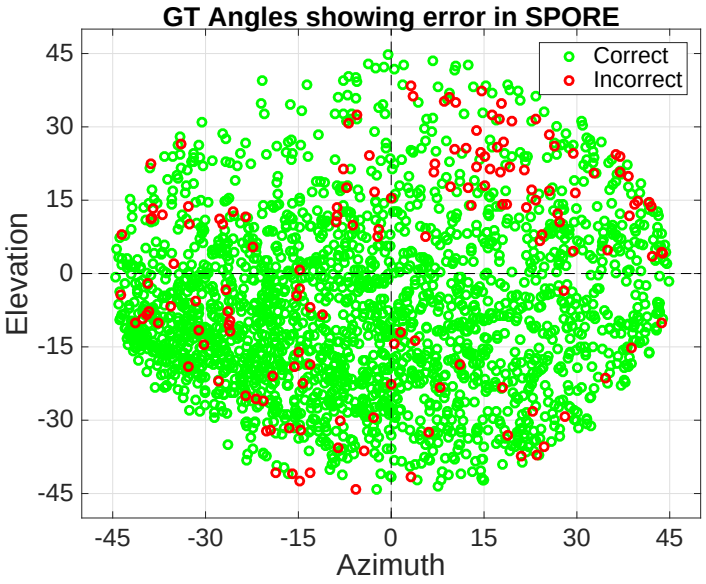
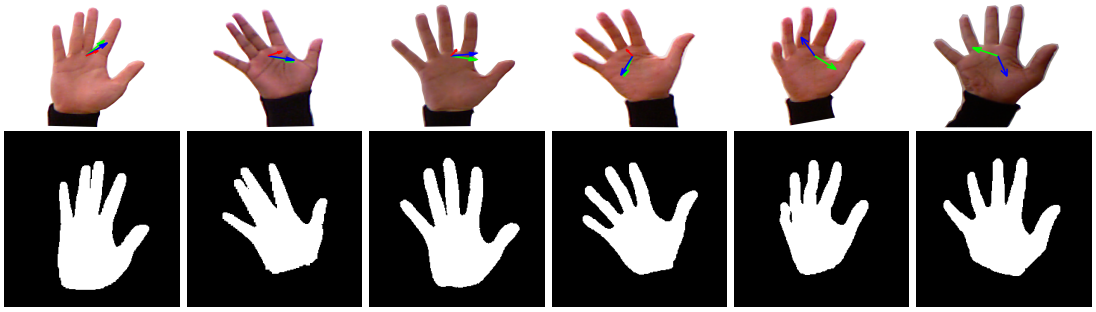


Fig. 6.8 GT angles showing regions with correct (green) and incorrect (red) orientation predictions using SPORE. We label a sample as being correctly (incorrectly) predicted if the absolute error is less (greater) than 15°.

increasing the number of stages does not contribute towards better accuracy. In fact, we note that when we increase N then the last stages receive insufficient data. In case of the above experiments with $N = 6$ stages, the last stage receives a total of 673 samples which is far less than the first stage where all training samples are used. Our findings suggest that more stages in SPORE can be employed in cases where a sufficiently large dataset is available. We also evaluate SPORE using *Dataset 1* and present our results in Table 6.1. The results indicate that a SPORE model trained using *Dataset 3* can generalize well to infer orientation angles on a different unseen data. Moreover, we note that *Dataset 1* was collected using unguided orientation movements, where a number of unwanted orientations might exist. These variations are the reason for the drop in performance of SPORE when testing using *Dataset 1*. Our experimental results show that SPORE produces the results with the least error, and a paired t-test with a p-value less than 0.05 demonstrates that SPORE's improvement over all other methods is statistically significant.

We also show the comparison of the proposed SPORE method with the most closely related ML-RF MtR method proposed in Chapter 5. In Fig. 6.7, we present the single-fold validation results showing the GT versus predicted plots for the proposed SPORE method



(a) (b) (c) (d) (e) (f)

Fig. 6.9 Success and failure cases for the proposed SPORE method. The GT orientation (green) and predicted orientation using SPORE (blue) and ML-RF MtR (red) are shown with arrows. (a)-(d) shows success cases where the proposed SPORE method is successfully able to infer the orientations. (e)-(f) shows the failure cases where the proposed method fails.

and the ML-RF MtR method. Fig. 6.7 (e)-(f) shows the comparison of both methods, where green arrows show predictions that were corrected using the proposed SPORE method and red arrows show the predictions that were incorrectly inferred by the proposed method. In these comparisons, we label predictions as correct when the absolute error is less than 15° and incorrect when it is greater than 15°. We observe that in this comparison 162 incorrectly inferred predictions by ML-RF MtR are corrected by the proposed SPORE method. This is due to the ability of our proposed SPORE method to define the latent variable space using predictions from previous stages. Such approach, however, is absent from the ML-RF MtR method where the latent variable space is explicitly defined based on the observation that the hand can be (i) fronto-parallel, or facing (ii) right, (iii) left, (iv) upwards or (v) downwards with respect to the camera. Moreover, we visualize these correct and incorrect predictions for SPORE using GT orientation angles in Fig. 6.8. It can be observed that most of the incorrect predictions lie in regions where lesser samples are captured due to unnatural hand movements required to acquire those orientations. Fig. 6.9 shows some cases for the proposed SPORE method. We observe that the proposed method fails on difficult samples where the fingers are not completely outstretched (Fig. 6.9 (e)-(f)). We also compare SPORE to ML-RF MtR K-means where the latent variables are defined using k-means for the ML-RF MtR method proposed in Chapter 5. This is to understand whether SPORE is learning complex latent variable-based representations or if it is only doing automatic clustering at

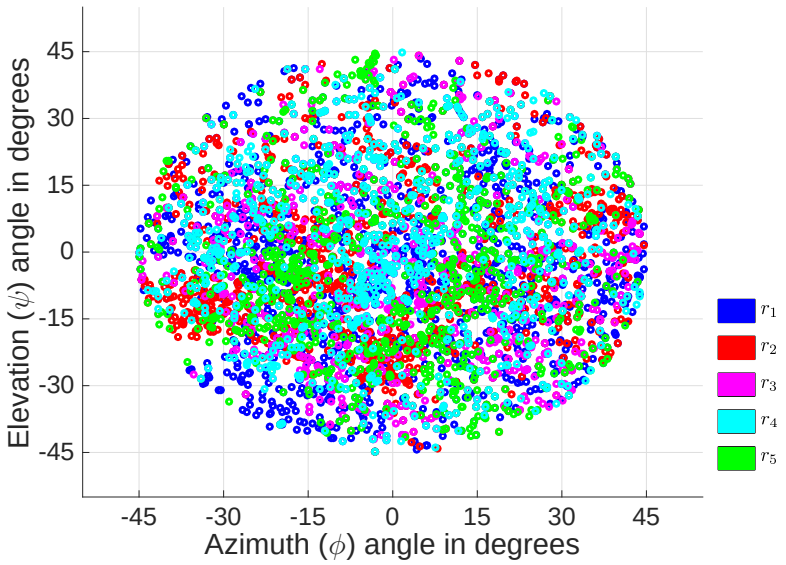


Fig. 6.10 Learned latent variables shown for different stages of SPORE. Each sample is shown with GT orientations angles and the learned latent variable representation shown in colors representing $r_n \in \{r_1, r_2, r_3 \cdots r_N\}$ for $N = 5$.

each stage with ML-RF MtR's marginalization through regression method. The results in Table 6.1 and Fig. 6.6 (a) indicate that indeed the learned representations are meaningful as they enable SPORE to outperform ML-RF MtR K-means. Fig. 6.10 shows the learned latent variable representation for SPORE. It can be seen that this representation builds a complex relationship between the learned expert regressors. While such relationship is not dictated by a trend, it does enable the SPORE method to outperform the comparison methods. Moreover, in Fig. 6.11 we present the easy versus harder to learn hand orientation samples. In Fig. 6.11 (a) easy samples are presented that the SPORE learned from the first stage. Fig. 6.11 (b) shows harder to train samples that are used for learning the next stages of SPORE. It can be seen that easy samples contain limited inter-person variation in hand shape, size and style, whereas harder samples have additional variations induced due to the movement of fingers, affecting the inter-finger spacing.

6.2.4 Experimental Validation using Multiple Pose Dataset

We use the *Dataset 4* to evaluate the extension of our proposed SPORE method for inferring both hand orientation and pose simultaneously. The MAE in degrees for the single-fold

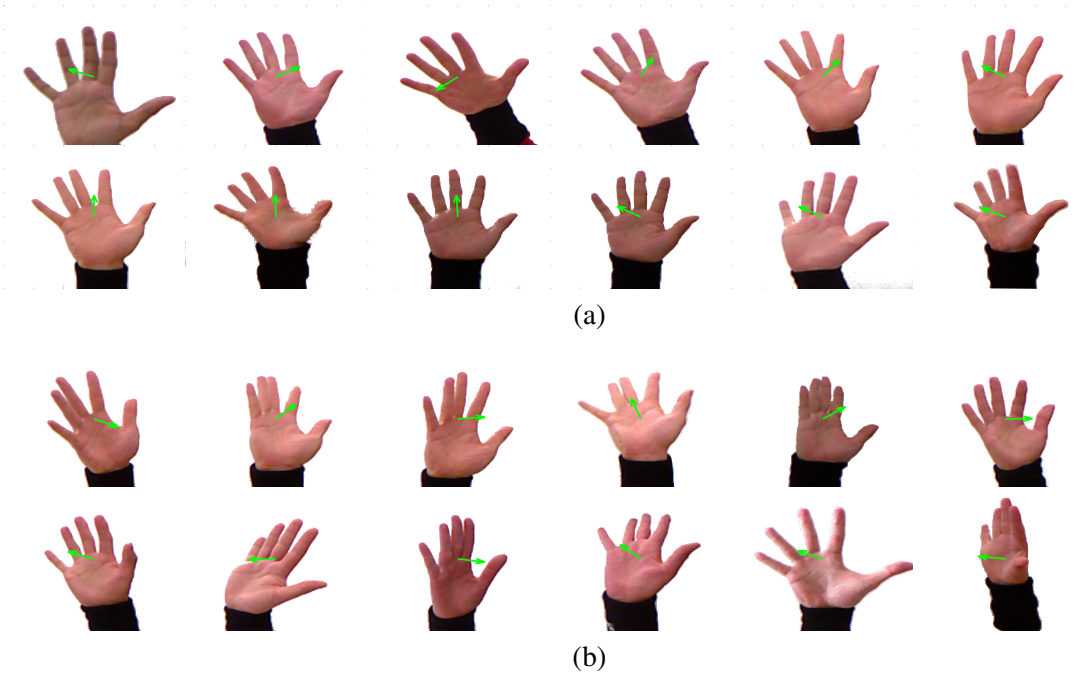


Fig. 6.11 Easy versus hard training samples for SPORE. (a) shows easy training samples that are successfully learned in the first stage regressor with error $|\mathbf{o}_p(k) - \mathbf{o}_k| < \sigma$. (b) shows harder training samples, with error $|\mathbf{o}_p(k) - \mathbf{o}_k| > \sigma$, that are not completely learned by the first stage regressor and hence are selected for the next training stage. Green arrows show the GT orientations. The difference between easy and hard samples can be seen in terms of inter-person pose, shape and style variations.

evaluation using this extension and the comparison methods is presented in Table 6.2. Furthermore, the percentage data under a given error (e) is shown in Fig. 6.6 (b). We notice that again, the proposed SPORE outperforms the comparison methods that infer hand orientation and pose simultaneously.

Furthermore, we compare the pose classification accuracy of the proposed SPORE method with RF Clf that learns only the pose classification [86]. We present confusion matrices for these results in Tables 6.3 and 6.4, respectively. It can be seen that SPORE outperforms the RF Clf for the pose classification task. This is due to the presence of the additional orientation information that SPORE uses to learn both hand orientation and pose simultaneously. The comparison RF Clf method lacks the orientation information, which is why it is unable to differentiate the poses with variations in orientation. In Fig. 6.12, we present the samples that are misclassified by RF Clf due to absence of orientation information. Moreover, the

Method used	**Azimuth** (ϕ)	**Elevation** (ψ)
SPORE	**8.72**°	**8.28**°
ML-RF MtR	9.62°	9.80°
ML-RF MtR K-means	9.75°	9.42°
SL-RF SV	14.53°	14.13°
RF Adaboost [121]	11.87°	11.08°
ML-RF1	10.61°	10.86°
ML-RF2	12.69°	11.65°
ML-RF3	10.80°	10.40°

Table 6.2 MAE in degrees for multiple pose experimental validation in Section 6.2.4.

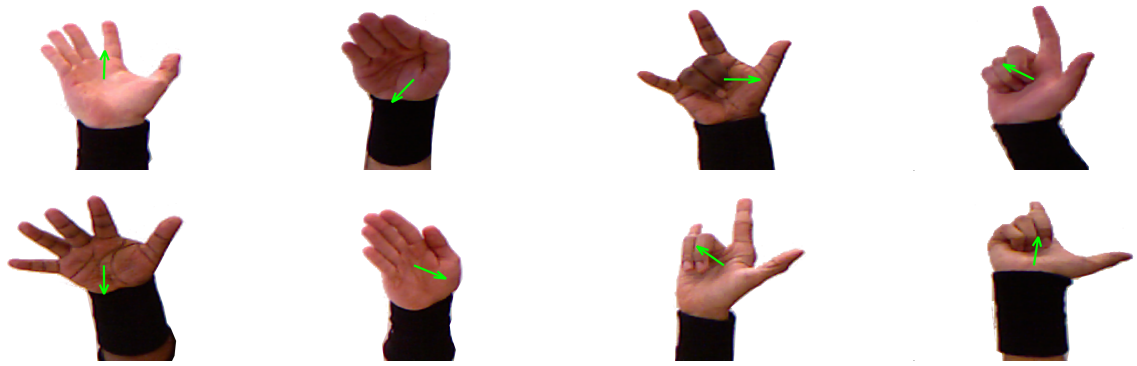
RF Clf method confuses pose χ_3 and χ_4. This is because under certain viewpoints hand undergoes self occlusion where the visible shape for χ_3 and χ_4 hand poses look similar (see Fig. 6.12 (c)). As the RF Clf does not learn from hand orientation, therefore it is not able to differentiate well between these poses. These results let us understand the importance of hand orientation in hand pose classification in 2D images. We observe that with the absence of such orientation information the classifier has difficulty in classifying poses under varying viewpoint.

Table 6.3 Hand pose classification results using SPORE.

		Predicted Pose			
		χ_1	χ_2	χ_3	χ_4
GT Pose	χ_1	97.94%	0.00%	1.90%	0.16%
	χ_2	0.00%	99.66%	0.17%	0.17%
	χ_3	0.15%	0.00%	98.97%	0.89%
	χ_4	0.00%	0.56%	1.50%	97.89%

Table 6.4 Hand pose classification results using RF Clf.

		Predicted Pose			
		χ_1	χ_2	χ_3	χ_4
GT Pose	χ_1	95.40%	0.00%	4.60%	0.00%
	χ_2	0.00%	94.16%	5.84%	0.00%
	χ_3	0.15%	0.00%	98.97%	0.89%
	χ_4	0.00%	1.54%	17.84%	80.62%



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

Fig. 6.12 Hand poses that are correctly inferred by the proposed SPORE method but misclassified by RF Clf. (a) shows χ_1 poses incorrectly classified as χ_3, (b) shows χ_2 pose incorrectly classified as χ_3, (c) shows χ_3 poses incorrectly classified as χ_4 and (d) shows χ_4 incorrectly classified as χ_3 by the RF Clf comparison method. Green arrows show the GT orientation information that is used by SPORE to correctly infer the hand pose. The orientation information is not used for training RF Clf.

6.2.5 Applications

To show the usefulness of the inferred hand orientation and pose, we present two application scenarios that utilize the inferred output in novel ways. Firstly, we extend the egocentric application scenario presented in Chapter 5 by introducing interaction through different hand postures. The video sequence captured from a head mounted camera contains a number of variations that include radial lens distortion and abrupt hand and head movements. The inferred orientation along with the location of the hand is used to render an augmented character on the hand palm. A user can interact with this character using different hand orientations and poses. In our application, χ_2 is used to trigger character spinning animation, χ_3 makes the character jump and χ_4 increases the scale of the character. Fig. 6.13 shows some frames from this Augmented Reality application of our method. We note that such interaction enables users to use their existing knowledge of how they manipulate physical objects and apply that in Augmented Reality, resulting in a natural interaction experience. Future work can be done to evaluate and compare our new interaction experience with

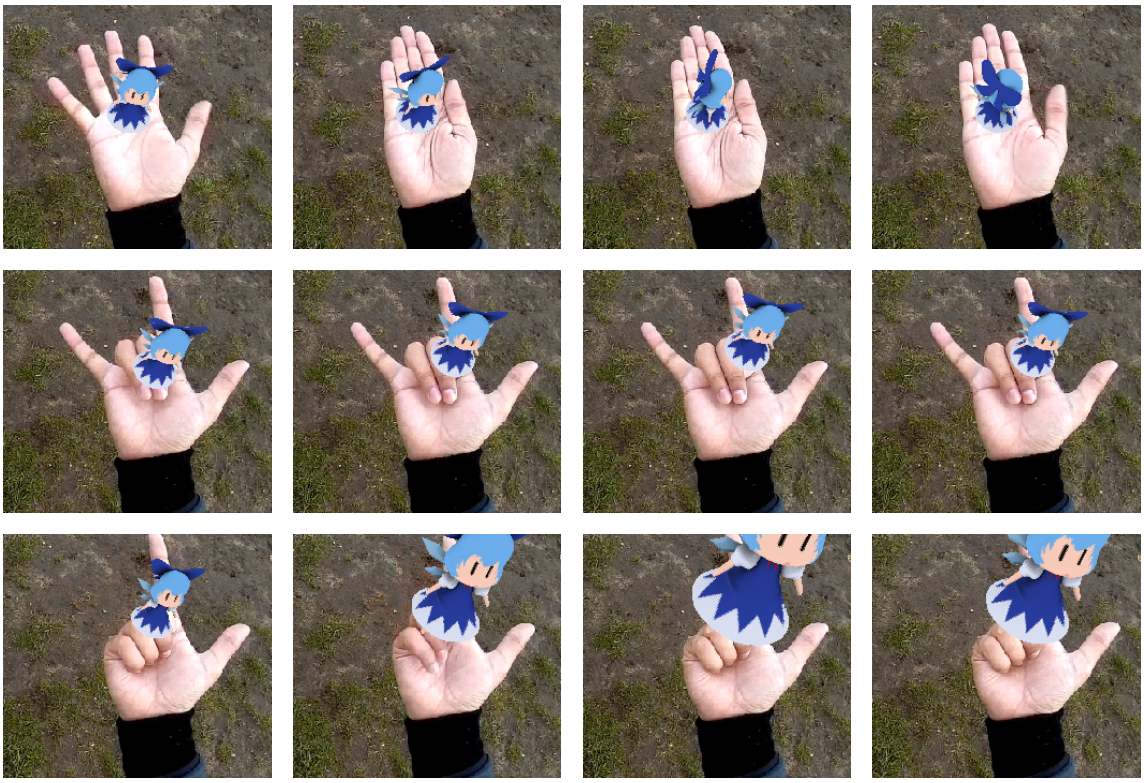
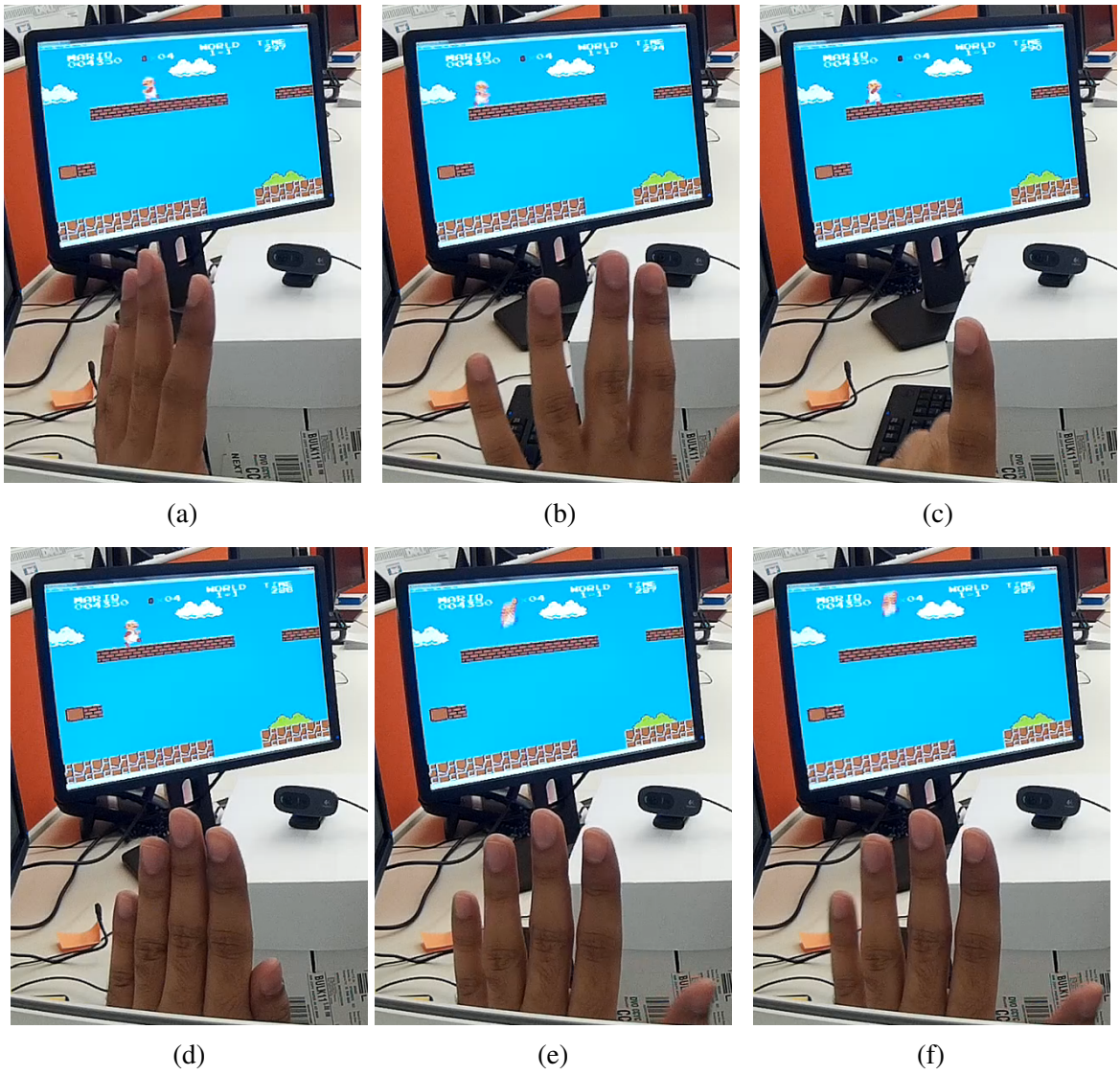


Fig. 6.13 Augmented Reality application in an egocentric setting. The hand orientation and pose from our proposed method can be used for interaction with augmented characters on the hand palm.

existing technologies, such as keyboards, mice and touch screen, that enable interaction with digital content.

To show the versatility of our proposed hand orientation and pose inference framework we also present a game interaction application (shown in Fig. 6.14). This demo utilized a standard off-the-shelf uncalibrated webcam. We utilized SPORE to infer both the hand orientation and pose which were used to control the movement and actions of a character in the game. The extracted orientations from the hand enable the user to control the direction of motion of the character, whereas the different hand postures trigger different actions like jump and fire. We use χ_2 to trigger jump and χ_3 to enable fire. This game provides some ideas of how our method can be used within existing applications and games.



(a) (b) (c)

(d) (e) (f)

Fig. 6.14 Game interaction application using the inferred hand orientation and pose. The game character can be moved using left and right orientation, whereas different hand poses trigger fire and jump actions. (a) and (b) show left and right movement using hand orientations, (c) shows the fire action performed using χ_3 pose and (d)-(f) show jump action performed using χ_2 pose. We use a standard off-the-shelf uncalibrated webcam to run this demo.

6.3 Summary

We proposed a staged probabilistic regression method that is capable of learning well from a number of variations within a dataset. The proposed method is based on multi-layered Random Forest, where the first layer consisted of a single marginalization weights regressor

and second layer contained an ensemble of expert learners. The expert learners are trained in stages, where each stage involved training and adding an expert learner to the intermediate model. After every stage, the intermediate model was evaluated to reveal a latent variable space defining a subset that the model had difficulty in learning from. This subset was used to train the next expert regressor. The posterior probabilities for each training sample were extracted from each expert regressors. These posterior probabilities were then used along with a Kullback-Leibler divergence-based optimization method to estimate the marginalization weights for each regressor. A marginalization weights regressor was trained using CDF and the estimated marginalization weights. We showed the extension of our work for simultaneous hand orientation and pose inference. The proposed method outperformed the state-of-the-art for marginalization of multi-layered Random Forest and hand orientation inference. Furthermore, we show that a method which simultaneously learns from hand orientation and pose outperforms pose classification as it is able to better understand the variations in pose induced due to viewpoint changes.

Chapter 7

Conclusions

7.1 Summary

A detailed review of the literature on hand pose estimation showed that the existing methods took either a generative or discriminative approach. The generative methods optimized a detailed hand model for estimating hand pose and were found to be computationally expensive to work on a mobile device. Moreover, most of these methods required an initialization step involving a fixed hand orientation and pose. In contrast, the recent developments in discriminative methods have shown the potential of a learned machine learning model for generalizing variations in a given dataset [3]. However, these methods were based on depth images which are not widely available on mobile devices due to consideration of power consumption, cost and form-factor [21]. Furthermore, it was found that despite using depth images, most existing datasets overlooked hand orientation [3]. These resulting methods worked for inferring complex hand pose articulation under limited viewpoint variations [34]. The introduction of egocentric devices requires methods that are robust to the changing viewpoint of different hand postures. As we have shown in this dissertation, the hand orientation information is important for learning discriminative models that can accurately infer hand pose under varying viewpoints. To this end, our work dealt with the problem of inferring hand orientation angles from 2D silhouette images of the hand. We describe our achievements below.

We proposed a unified framework for hand orientation regression that was further used to propose the single-layered and multi-layered Random Forest regression frameworks. We contributed a single-layered single-variate Random Forest (SL-RF SV) regression method that learned the mapping of Contour Distance Features (CDF) onto the hand orientation angles [28]. This framework presented a method for extracting CDF that encoded the shape variations in a planar hand pose for varying viewpoint. We also proposed a method for ground truth (GT) hand orientation annotation using depth images. To the best of our knowledge, the proposed method is the first to present automatic hand orientation annotation. SL-RF SV was evaluated on a dataset of varying orientation captured using a planar open hand pose from 13 participants. This method showed promising results while revealing some challenges that defined our later research work. SL-RF SV also showed potential novel application scenarios for using the inferred hand orientation for interaction in an Augmented Reality application. The dataset collection method used in this work was unguided, where we relied on participants' understanding of hand orientation variations. This resulted in a smaller dataset with many inconsistencies, motivating a need for the guided data collection method. Furthermore, a single model proved limited for learning the variation of hand orientation from the dataset, which was evident from the bias in the learned models.

We further contributed a method for multi-layered Random Forest Marginalization through Regression (ML-RF MtR). This method utilized the idea of dividing a complex learning task into smaller subsets and learning using multiple expert regressors [17, 21]. ML-RF MtR was composed of two layers, namely, marginalization weights and expert regression layer. Training of ML-RF MtR involved using a latent variable space to define subsets, with a simple observation that the hand can be oriented (i) fronto-parallel or facing (ii) right, (iii) left, (iv) upwards or (v) downwards with respect to the camera. Expert regressors were trained on the subsets defined by the latent variable space. The posterior probabilities corresponding to each sample in the training set were acquired from each of the trained expert regressors. Our main contribution came from the use of a marginalization weights regressor that learned the mapping of CDF to marginalization weights. We derived and applied a Kullback-Leibler divergence-based optimization method that estimated the marginalization weights

for the training data. The online prediction involved inferring the posterior probabilities from each expert regressor and the corresponding marginalization weights inferred using marginalization weights regressor. These probabilities were marginalized to infer the hand orientation angles. ML-RF MtR outperformed the state-of-the-art for both hand orientation and multi-layered Random Forest (ML-RF) marginalization methods. We also proposed a method for collecting the hand orientation dataset by guiding participants in a painting game application. This method enabled collection of consistent datasets, where more samples were acquired from each participant. The objective of the painting application was to give real-time feedback to participants so that they could perform variations to fully fill the hand orientation space. We also showed an egocentric Augmented Reality application scenario, where an uncalibrated head mounted camera was used to acquire color images of the hand. ML-RF MtR was used to infer the hand orientation angles, which were used to manipulate the orientation of an augmented character.

One limitation of ML-RF MtR was that it required explicit definition of latent variable space for defining subsets. This limited its applicability to a wider range of applications, as an explicit definition of the subsets is not always available. Furthermore, dividing subsets for cases where limited data is available could potentially result in fewer samples in each subset, hence producing under fitting models. To address this, we proposed a staged probabilistic regression (SPORE) method. This method utilized a staged learning approach where after training each stage, the evaluation of the intermediate model was used to extract the latent variable space for next stage. SPORE utilized the marginalization through regression proposed for ML-RF MtR that combined predictions from multiple stages. It outperformed SL-RF SV, ML-RF MtR and a number of comparison methods including Random Forest with Adaboost and ML-RF marginalization methods. We extended SPORE to simultaneously infer hand orientation and pose, which showed the significance of hand orientation in learning models that better understand hand pose under varying viewpoints. Comparison of this extension with Random Forest classification method that only learned from hand pose showed that indeed a model that simultaneously learns from hand orientation and pose can outperform a pose only model. An egocentric Augmented Reality application scenario

was presented, where the inferred hand orientations and poses were used to manipulate and interact with an augmented character. We also described how our method can be used to enable hand-based game interactions. In this application, the left/right movements of the game character were controlled using the hand orientations, whereas the hand poses were used to trigger different actions, such as fire and jump. Both application scenarios utilized uncalibrated off-the-shelf cameras.

7.2 Limitations

The human hand exhibits high Degrees-Of-Freedom (DoF) resulting in a number of different hand postures. The methods proposed in this dissertation followed a number of assumptions that introduce a number of limitations. We discuss these limitations below.

The hand poses were assumed to be planar in order to help with the acquisition of GT hand orientation datasets. This limits the application of these methods to a wider range of applications, where complex hand-based interactions are desired. Moreover, the orientation angles were limited to the range $\sqrt{\phi^2 + \psi^2} \leq 45^\circ$ which also imposed restrictions on the application scenario. Although such orientations are useful, there exist a number of viewpoints in the egocentric scenario that fall outside the defined range. Furthermore, the color images from 2D monocular cameras were assumed to have an uncluttered background, which aided in a good segmentation of the hands. However, such assumption limits the applicability of the proposed methods to the dynamically changing egocentric scenarios. The segmentations of the wrist were aided by using a black sweatband, which is not present in most real-world scenarios. This can be addressed by adding a method that detects the ulnar points on the wrist and uses them to segment out the arm [47].

The problem of inferring 3D hand orientations from 2D monocular images is ill-posed, where a single hand shape can be obtained from a number of different orientations. This results in the symmetry problem, where the proposed model infers symmetrically opposite hand orientations for a given hand shape. Moreover, the employed Contour Distance Features represent hand shape information only, where the additional shading and texture information

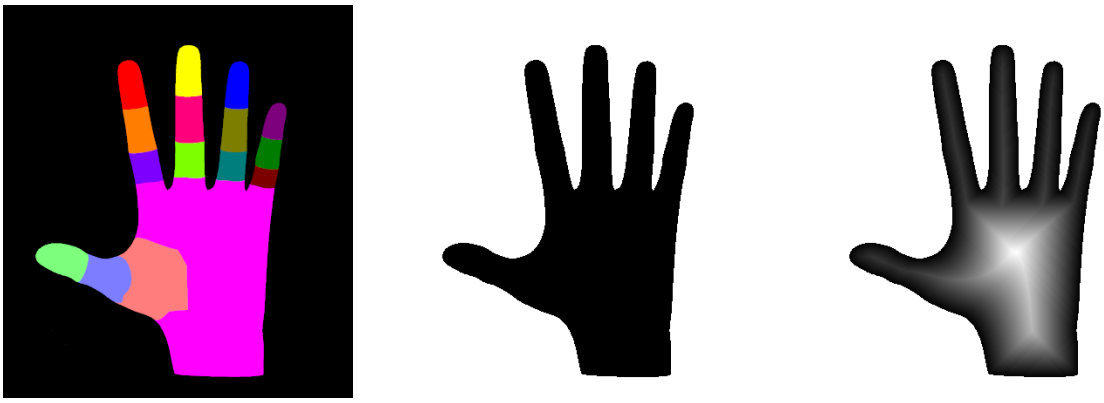
might be useful for resolving ambiguity in the hand orientation problem. However, using shading and texture might pose additional challenges such as generalizing across a range of different lighting conditions and skin tones. While we tested our method using different cameras, the experimental validation utilized images from only one camera. Although this enabled an in-depth analysis of our methods, future work can utilize different cameras to further validate the performance. Furthermore, the proposed methods utilized only a single hand, where egocentric scenarios may contain two handed interactions. This can be addressed in future work by presenting a model that simultaneously learns to classify the hand into the left or the right hand, infers the orientation and pose. The accuracy of the proposed hand orientation inference methods is influenced by symmetry problem resulting in abrupt changes in the application scenario. This can be addressed by exploiting the temporal coherence using an existing tracking technique [120].

The color and depth images, from Microsoft Kinect v1, that were employed in our work also impose some limitations. The resolution of these images is 640x480 pixels, where hand normally occupies about 150x150 pixels. This relatively small size makes the extracted GT orientation angles susceptible to quantization errors and noise from the depth images [122]. Moreover, at this resolution, the depth information for fingertips and fingers pointing towards the camera is not fully captured. Nevertheless, the planar hand assumption along with RANSAC-based GT orientation extraction makes our proposed methods tolerate these noise and errors. If, in future work, non-planar hand poses are used then the noise in depth images can raise issues in GT orientation angle estimation. These limitations can be addressed by utilizing our data capture as well as hand orientation and pose estimation frameworks with newer versions of depth sensors, such as Microsoft Kinect v2.

We now describe our future work, which aims at addressing these limitations, in the next section.

7.3 Future Work

Below we discuss the future directions of the work proposed in this dissertation



(a) (b) (c)

Fig. 7.1 Hand part label and distance transform-based features showing (a) labeled synthetic image rendered in Poser[1], the corresponding (b) silhouette and (c) distance transform.

Dataset for Multiple Articulated Hand Pose Future work can extend the painting game-based hand orientation dataset collection method for acquiring dataset of articulated hand pose with varying orientations. This can be achieved by utilizes additional sensors such as an Inertial Measurement Unit (IMU) placed on the palm. When registered with the 2D monocular sensors, viewpoint variations for nonplanar hand pose can be captured. These new datasets can be used with our proposed SPORE method to infer both hand orientation and pose from 2D monocular images.

Feature Extraction While Contour Distance Features proved to be efficient for hand orientation regression from planar hand pose, they provide limited information for the inferring complex hand articulations. Our future work will aim to propose more effective and detailed feature extraction techniques from 2D color images. In particular, we envision using the distance transform of silhouette images to propose features, similar to [15, 17, 56]. Fig. 7.1 shows a labeled synthetic hand image along with the corresponding silhouette image and the extracted distance transform features. As it is evident from Fig. 7.1 (c), the distance transform features represent the details of both the hand and the parts within the hand. These features will enable us to build an approach based on different parts of the hand, which can then be used to distinguish different articulations and viewpoints. Part-based features will also help us make a

generalized approach, as specific parts of the hand are more likely to be similar for different persons as compared to the whole hand in the Contour Distance Feature [15]. Moreover, features based on texture and illumination can be utilized to address the symmetry problem.

Temporal Coherence A number of post-processing methods can be used to exploit the temporal coherence in a sequence of hand orientation estimation. This could significantly improve the performance of the proposed methods as in most failure cases the symmetrically opposite hand orientations are inferred. However, such errors are short-lived, hence utilizing a filtering method is an interesting future direction that can be explored. A dynamical system model can be used to infer the internal states such as velocity and acceleration of an object by using observable states such as distance moved [71, 120]. When applying such models to filter the inferred hand orientations, changes in orientation angles can be used to predict the angular velocity and angular acceleration. Once the internal states of the hand are known, one can reason and reject possible erroneous orientation angles resulting from the symmetry problem.

Hybrid Methods A major limitation of generative methods is their inability to recover from tracking errors. The inferred hand orientation and pose from our proposed methods can be used as a re-initializer for a generative model-based articulated hand pose estimation. Moreover, it can help improve the computational efficiency as well as the accuracy of such methods by proposing a hand orientation and pose that is close to the final solution. Another limitation of generative methods is their requirement for initializing hand orientation and pose. Our discriminative hand orientation and pose estimation method can address this limitation by allowing the model to work with a number of different poses with varying orientations.

Generalizing for Other Learning Problems This dissertation focused on using SPORE for hand orientation and pose inference. We note that the proposed method can be generalized to other domains. SPORE can be used with any probabilistic regressor or

[1]Poser Pro 2014: http://my.smithmicro.com/poser-pro-2014.html

classifier, where the dataset contains large variations that are not fully learned with a single model. One similar problem, where our method can be applied to, is the head orientation estimation from 2D monocular images [77]. A trained SPORE model for this problem can be useful in CCTV surveillance applications, where the orientation of a person's head can be used to help determine the higher level activities.

Real-world Applications The proposed hand orientation and pose estimation method can be deployed to latest Augmented and Virtual Reality platforms such as Microsoft Hololens[2] and Oculus Rift[3]. Our method can provide simple interaction and manipulation of virtual objects that can enhance the user experience on such devices. Possible extensions to our work can include a larger vocabulary of hand poses and orientations, which can further contribute to a richer interaction experience.

Exploring Other Latent Variables This dissertation aims to learn and utilize latent variables for hand orientation and pose inference only. The proposed methods can be extended to explore other relevant latent variables such as the size of the hand, categorized by height and width, freedom of movement of different joints, finger separation area etc. As we have shown in this dissertation, learning such latent variables can significantly contribute towards improvement in overall accuracy of a given machine learning model.

Extension to Two Hands The proposed dataset capture as well as hand orientation and pose estimation frameworks can be extended to two-handed interaction scenarios. Given a hand orientation and pose dataset captured from two hands, the proposed SPORE method can be trained to simultaneously infer handedness, orientation and pose. This can be used to identify the hand and provide appropriate interaction using orientation and pose.

Overlapping Latent Variable Boundaries for ML-RF MtR Our proposed work for SPORE showed that overlapping boundaries for latent variable definition can be advantageous.

[2]https://www.microsoft.com/microsoft-hololens/en-gb
[3]https://www3.oculus.com/en-us/rift/

The proposed ML-RF MtR only uses non-overlapping boundaries. Therefore, our future work may involve exploring the use of explicit definition of overlapping boundaries to see if ML-RF MtR can surpass SPORE in performance.

References

[1] Alejandro Betancourt, Pietro Morerio, Carlo S Regazzoni, and Matthias Rauterberg. The evolution of first person vision methods: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(5):744–760, 2015.

[2] Ali Erol, George Bebis, Mircea Nicolescu, Richard D Boyle, and Xander Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, 108(1):52–73, 2007.

[3] James S Supancic, Grégory Rogez, Yi Yang, Jamie Shotton, and Deva Ramanan. Depth-based hand pose estimation: data, methods, and challenges. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1868–1876, 2015.

[4] Bruce H. Thomas. Recent trends in augmented and virtual reality. Online, January 2016. Accessed on: 08/11/2016.

[5] Stephen Cass and CQ Choi. Google glass, hololens, and the real future of augmented reality. *IEEE Spectrum*, 14, 2015.

[6] Ken Perlin. Future reality: How emerging technologies will change language itself. *IEEE Computer Graphics and Applications*, 36(3):84–89, 2016.

[7] Greg Kipper and Joseph Rampolla. *Augmented Reality: an emerging technologies guide to AR*. Elsevier, 2012.

[8] Frank Biocca and Mark R Levy. *Communication in the age of virtual reality*. Routledge, 2013.

[9] Mark Billinghurst and Thad Starner. Wearable devices: new ways to manage information. *Computer*, 32(1):57–64, 1999.

[10] Dan Wang, Zheng Xiang, and Daniel R Fesenmaier. Smartphone use in everyday life and travel. *Journal of Travel Research*, 55(1):52–63, 2016.

[11] Javier Hernandez and Rosalind W Picard. Senseglass: using google glass to sense daily emotions. In *Proceedings of the adjunct publication of the 27th annual ACM symposium on User interface software and technology*, pages 77–78. ACM, 2014.

[12] Reza Rawassizadeh, Blaine A Price, and Marian Petre. Wearables: has the age of smartwatches finally arrived? *Communications of the ACM*, 58(1):45–47, 2015.

[13] Alejandro Betancourt, Pietro Morerio, Emilia Barakova, Lucio Marcenaro, Matthias Rauterberg, and Carlo Regazzoni. Left/right hand segmentation in egocentric videos. *Computer Vision and Image Understanding*, 2016.

[14] Jungong Han, Ling Shao, Dong Xu, and Jamie Shotton. Enhanced computer vision with microsoft kinect sensor: A review. *Cybernetics, IEEE Transactions on*, 43(5):pp.1318–1334, 2013.

[15] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

[16] Jamie Shotton, Ross Girshick, Andrew Fitzgibbon, Toby Sharp, Mat Cook, Mark Finocchio, Richard Moore, Pushmeet Kohli, Antonio Criminisi, Alex Kipman, et al. Efficient human pose estimation from single depth images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2821–2840, 2013.

[17] Cem Keskin, Furkan Kıraç, Yunus Emre Kara, and Lale Akarun. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *Computer Vision–ECCV 2012*, pages 852–863. Springer, 2012.

[18] Danhang Tang, Tsz-Ho Yu, and Tae-Kyun Kim. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *2013 IEEE International Conference on Computer Vision (ICCV)*, pages 3224–3231. IEEE, 2013.

[19] Iasonas Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *2011 IEEE International Conference on Computer Vision (ICCV)*, pages 2088–2095. IEEE, 2011.

[20] Jonathan Taylor, Lucas Bordeaux, Thomas Cashman, Bob Corish, Cem Keskin, Toby Sharp, Eduardo Soto, David Sweeney, Julien Valentin, Benjamin Luff, et al. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM Transactions on Graphics (TOG)*, 35(4):143, 2016.

[21] Sean Ryan Fanello, Cem Keskin, Shahram Izadi, Pushmeet Kohli, David Kim, David Sweeney, Antonio Criminisi, Jamie Shotton, Sing Bing Kang, and Tim Paek. Learning to be a depth camera for close-range human capture and interaction. *ACM Transactions on Graphics (TOG)*, 33(4):86, 2014.

[22] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *BMVC*, volume 1, page 3, 2011.

[23] Ying Wu, John Lin, and Thomas S Huang. Analyzing and capturing articulated hand motion in image sequences. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(12):1910–1922, 2005.

[24] Robert Y Wang and Jovan Popović. Real-time hand-tracking with a color glove. In *ACM Transactions on Graphics (TOG)*, volume 28, page 63. ACM, 2009.

[25] Taehee Lee and Tobias Hollerer. Handy ar: Markerless inspection of augmented reality objects using fingertip tracking. In *11th IEEE International Symposium on Wearable Computers, 2007*, pages 83–90. IEEE, 2007.

[26] Martin de La Gorce, David J Fleet, and Nikos Paragios. Model-based 3d hand pose estimation from monocular video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1793–1805, 2011.

[27] Martin de La Gorce and Nikos Paragios. A variational approach to monocular hand-pose estimation. *Computer Vision and Image Understanding*, 114(3):363–372, 2010.

[28] Muhammad Asad and Greg Slabaugh. Hand orientation regression using random forest for augmented reality. In *International Conference on Augmented and Virtual Reality*, 2014.

[29] Andrew K Palmer, Frederick W Werner, Dennis Murphy, and Richard Glisson. Functional wrist motion: a biomechanical study. *The Journal of hand surgery*, 10(1):39–46, 1985.

[30] Muhammad Asad and Greg Slabaugh. Learning marginalization through regression for hand orientation inference. In *Computer Vision and Pattern Recognition (CVPR) Second Workshop on Observing and Understanding Hands in Action (HANDS)*, 2016.

[31] Muhammad Asad and Greg Slabaugh. Spore: Staged probabilistic regression for hand orientation and pose inference. *Computer Vision and Image Understanding (Under Review)*, 2016.

[32] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, et al. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3633–3642. ACM, 2015.

[33] Chi Xu and Li Cheng. Efficient hand pose estimation from a single depth image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3456–3462, 2013.

[34] Xiao Sun, Yichen Wei, Shuang Liang, Xiaoou Tang, and Jian Sun. Cascaded hand pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 824–832, 2015.

[35] Yoshiaki Mizuchi, Yoshinobu Hagiwara, Akimasa Suzuki, Hiroki Imamura, and Yongwoon Choi. Monocular 3d palm posture estimation based on feature-points robust against finger motion. In *2013 13th International Conference on Control, Automation and Systems (ICCAS)*, pages 1014–1019. IEEE, 2013.

[36] Bjoern Stenger, Paulo RS Mendonça, and Roberto Cipolla. Model-based 3d tracking of an articulated hand. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–310. IEEE, 2001.

[37] Ying Wu, John Y Lin, and Thomas S Huang. Capturing natural hand articulation. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 426–432. IEEE, 2001.

[38] Iason Oikonomidis, Manolis IA Lourakis, and Antonis A Argyros. Evolutionary quasi-random search for hand articulations tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3422–3429, 2014.

[39] Luca Ballan, Aparna Taneja, Jürgen Gall, Luc Van Gool, and Marc Pollefeys. Motion capture of hands in action using discriminative salient points. In *European Conference on Computer Vision*, pages 640–653. Springer, 2012.

[40] Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. Realtime and robust hand tracking from depth. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1106–1113, 2014.

[41] Srinath Sridhar, Antti Oulasvirta, and Christian Theobalt. Interactive markerless articulated hand motion tracking using rgb and depth data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2456–2463, 2013.

[42] Stan Melax, Leonid Keselman, and Sterling Orsten. Dynamics based 3d skeletal hand tracking. In *Proceedings of Graphics Interface 2013*, pages 63–70. Canadian Information Processing Society, 2013.

[43] Sameh Khamis, Jonathan Taylor, Jamie Shotton, Cem Keskin, Shahram Izadi, and Andrew Fitzgibbon. Learning an efficient model of hand shape variation from depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2540–2548, 2015.

[44] David Joseph Tan, Thomas Cashman, Jonathan Taylor, Andrew Fitzgibbon, Daniel Tarlow, Sameh Khamis, Shahram Izadi, and Jamie Shotton. Fits like a glove: Rapid and reliable hand shape personalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[45] Iasonas Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Markerless and efficient 26-dof hand pose recovery. In *Asian Conference on Computer Vision*, pages 744–757. Springer, 2010.

[46] Jonathan Taylor, Richard Stebbing, Varun Ramakrishna, Cem Keskin, Jamie Shotton, Shahram Izadi, Aaron Hertzmann, and Andrew Fitzgibbon. User-specific hand modeling from monocular depth sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 644–651, 2014.

[47] Taehyun Rhee, Ulrich Neumann, and John P Lewis. Human hand modeling from surface anatomy. In *Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 27–34. ACM, 2006.

[48] Muhammad Asad, Enguerrand Gentet, Rilwan Remilekun Basaru, and Greg Slabaugh. Generating a 3d hand model from frontal color and range scans. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 4589–4593. IEEE, 2015.

[49] Fred L Bookstein et al. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence*, 11(6):567–585, 1989.

[50] Ross Girshick, Jamie Shotton, Pushmeet Kohli, Antonio Criminisi, and Andrew Fitzgibbon. Efficient regression of general-activity human poses from depth images. In *2011 International Conference on Computer Vision*, pages 415–422. IEEE, 2011.

[51] Mao Ye, Qing Zhang, Liang Wang, Jiejie Zhu, Ruigang Yang, and Juergen Gall. A survey on human motion analysis from depth data. In *Time-of-flight and depth imaging. sensors, algorithms, and applications*, pages 149–187. Springer, 2013.

[52] Rómer Rosales and Stan Sclaroff. Combining generative and discriminative models in a framework for articulated pose estimation. *International Journal of Computer Vision*, 67(3):251–276, 2006.

[53] Toby Sharp. Implementing decision trees and forests on a gpu. In *European conference on computer vision*, pages 595–608. Springer, 2008.

[54] Brian Van Essen, Chris Macaraeg, Maya Gokhale, and Ryan Prenger. Accelerating a random forest classifier: Multi-core, gp-gpu, or fpga? In *Field-Programmable Custom Computing Machines (FCCM), 2012 IEEE 20th Annual International Symposium on*, pages 232–239. IEEE, 2012.

[55] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[56] Cem Keskin, Furkan Kıraç, Yunus Emre Kara, and Lale Akarun. Real time hand pose estimation using depth sensors. In *Consumer Depth Cameras for Computer Vision*, pages 119–137. Springer, 2013.

[57] Hui Liang, Junsong Yuan, and Daniel Thalmann. Parsing the hand in depth images. *IEEE Transactions on Multimedia*, 16(5):1241–1253, 2014.

[58] Danhang Tang, Hyung Jin Chang, Alykhan Tejani, and Tae-Kyun Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3786–3793, 2014.

[59] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (TOG)*, 33(5):169, 2014.

[60] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns.

[61] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Hands deep in deep learning for hand pose estimation. *arXiv preprint arXiv:1502.06807*, 2015.

[62] Markus Oberweger, Gernot Riegler, Paul Wohlhart, and Vincent Lepetit. Efficiently creating 3d training data for fine hand pose estimation. *arXiv preprint arXiv:1605.03389*, 2016.

[63] Georg Poier, Konstantinos Roditakis, Samuel Schulter, Damien Michel, Horst Bischof, and Antonis A Argyros. Hybrid one-shot 3d hand pose estimation by exploiting uncertainties. *arXiv preprint arXiv:1510.08039*, 2015.

[64] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Training a feedback loop for hand pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3316–3324, 2015.

[65] Eng-Jon Ong and Richard Bowden. A boosted classifier tree for hand shape detection. In *2004. Proceedings. Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 889–894. IEEE, 2004.

[66] Matthias Dantone, Juergen Gall, Gabriele Fanelli, and Luc Van Gool. Real-time facial feature detection using conditional regression forests. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2578–2585. IEEE, 2012.

[67] Min Sun, Pushmeet Kohli, and Jamie Shotton. Conditional regression forests for human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3394–3401. IEEE, 2012.

[68] Kah Pin Ng, Guat Yew Tan, and Ya Ping Wong. Vision-based hand detection for registration of virtual objects in augmented reality. *International Journal of Future Computer and Communication*, 2(5):423–427, 2013.

[69] Haruhisa Kato and Akio Yoneyama. A line-based palm-top detector for mobile augmented reality. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 208–211. ACM, 2012.

[70] Haruhisa Kato and Tsuneo Kato. A marker-less augmented reality based on fast fingertip detection for smart phones. In *2011 IEEE International Conference on Consumer Electronics (ICCE)*, pages 127–128. IEEE, 2011.

[71] Greg Welch and Gary Bishop. An introduction to the kalman filter. 2006. *University of North Carolina: Chapel Hill, North Carolina, US*, 2006.

[72] Hirokazu Kato. Artoolkit: library for vision-based augmented reality. *IEICE, PRMU*, 6:79–86, 2002.

[73] Dana H Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122, 1981.

[74] Joseph J Laviola. A comparison of unscented and extended kalman filtering for estimating quaternion motion. In *American Control Conference, 2003. Proceedings of the 2003*, volume 3, pages 2435–2440. IEEE, 2003.

[75] Erik Murphy-Chutorian and Mohan Manubhai Trivedi. Head pose estimation in computer vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 31(4):607–626, 2009.

[76] Gabriele Fanelli, Juergen Gall, and Luc Van Gool. Real time head pose estimation with random regression forests. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 617–624. IEEE, 2011.

[77] Vincent Drouard, Sileye Ba, Georgios Evangelidis, Antoine Deleforge, and Radu Horaud. Head pose estimation via probabilistic high-dimensional regression. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 4624–4628. IEEE, 2015.

[78] Liang Zhao, Gopal Pingali, and Ingrid Carlbom. Real-time head orientation estimation using neural networks. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I–297. IEEE, 2002.

[79] Thanarat Horprasert, Yaser Yacoob, and Larry S Davis. Computing 3d head orientation from a monocular image sequence. In *25th Annual AIPR Workshop on Emerging Applications of Computer Vision*, pages 244–252. International Society for Optics and Photonics, 1997.

[80] Yali Li, Shengjin Wang, and Xiaoqing Ding. Person-independent head pose estimation based on random forest regression. In *2010 IEEE International Conference on Image Processing*, pages 1521–1524. IEEE, 2010.

[81] Andrew Ng. Cs229 lecture notes. *CS229 Lecture notes*, 1(1):1–3, 2000.

[82] Gilles Louppe. Understanding random forests: From theory to practice. *arXiv preprint arXiv:1407.7502*, 2014.

[83] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2014.

[84] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[85] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[86] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[87] Christopher M Bishop. Pattern recognition. *Machine Learning*, 128, 2006.

[88] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. Book in preparation for MIT Press, 2016.

[89] Cheng Li and Kris M Kitani. Pixel-level hand detection in ego-centric videos. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3570–3577. IEEE, 2013.

[90] Michael J Jones and James M Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96, 2002.

[91] Vladimir Vezhnevets, Vassili Sazonov, and Alla Andreeva. A survey on pixel-based skin color detection techniques. In *Proc. Graphicon*, volume 3, pages 85–92. Moscow, Russia, 2003.

[92] Zhou Ren, Junsong Yuan, Jingjing Meng, and Zhengyou Zhang. Robust part-based hand gesture recognition using kinect sensor. *IEEE transactions on multimedia*, 15(5):1110–1120, 2013.

[93] Lawrence Y Deng, Jason C Hung, Huan-Chao Keh, Kun-Yi Lin, Yi-Jen Liu, and Nan-Ching Huang. Real-time hand gesture recognition by shape context based matching and cost matrix. *JNW*, 6(5):697–704, 2011.

[94] Peter Kontschieder, Madalina Fiterau, Antonio Criminisi, and Samuel Rota Bulo. Deep neural decision forests. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1467–1475, 2015.

[95] Erdem Yoruk, Ender Konukoglu, Bülent Sankur, and Jérôme Darbon. Shape-based hand recognition. *IEEE Transactions on Image Processing*, 15(7):1803–1815, 2006.

[96] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[97] Linda Shapiro and R Haralick. Computer and robot vision. *Reading: Addison-Wesley*, 8, 1992.

[98] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems. *J. Mach. Learn. Res*, 15(1):3133–3181, 2014.

[99] Antonio Criminisi and Jamie Shotton. *Decision forests for computer vision and medical image analysis*. Springer, 2013.

[100] Danhang Tang. *3D hand pose regression with variants of decision forests*. PhD thesis, Imperial College London, 2015.

[101] Rich Caruana, Nikos Karampatziakis, and Ainur Yessenalina. An empirical evaluation of supervised learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 96–103. ACM, 2008.

[102] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[103] Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.

[104] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[105] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[106] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

[107] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[108] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[109] Lindsay I Smith. A tutorial on principal components analysis. *Cornell University, USA*, 51:52, 2002.

[110] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995.

[111] Anand Rajaraman and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2011.

[112] Sasan Karamizadeh, Shahidan M Abdullah, Azizah A Manaf, Mazdak Zamani, and Alireza Hooman. An overview of principal component analysis. *Journal of Signal and Information Processing*, 4(3B):173, 2013.

[113] Michael E Wall, Andreas Rechtsteiner, and Luis M Rocha. Singular value decomposition and principal component analysis. In *A practical approach to microarray data analysis*, pages 91–109. Springer, 2003.

[114] Michael E Leventon, W Eric L Grimson, and Olivier Faugeras. Statistical shape influence in geodesic active contours. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 316–323. IEEE, 2000.

[115] Guoqiang Peter Zhang. Neural networks for classification: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 30(4):451–462, 2000.

[116] Leo Breiman. Using adaptive bagging to debias regressions. Technical report, Technical Report 547, Statistics Dept. UCB, 1999.

[117] Guoyi Zhang and Yan Lu. Bias-corrected random forests in regression. *Journal of Applied Statistics*, 39(1):151–160, 2012.

[118] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[119] S. Hallman and C. C. Fowlkes. Oriented edge forests for boundary detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[120] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2):174–188, 2002.

[121] Dimitri P Solomatine and Durga L Shrestha. Adaboost. rt: a boosting algorithm for regression problems. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 2, pages 1163–1168. IEEE, 2004.

[122] Muhammad Asad and Charith Abhayaratne. Kinect depth stream pre-processing for hand gesture recognition. In *2013 IEEE International Conference on Image Processing*, pages 3735–3739. IEEE, 2013.