SPORE: Staged Probabilistic Regression for Hand Orientation Inference

Muhammad Asad and Greg Slabaugh

Department of Computer Science, City University London, UK

Abstract

Learning the global hand orientation from 2D monocular images is a challenging task, as the projected hand shape is affected by a number of variations. These include inter-person hand shape and size variations, intra-person pose and style variations and self-occlusion due to varying hand orientation. Given a hand orientation dataset containing these variations, a single regressor proves to be limited for learning the mapping of hand silhouette images onto the orientation angles. We address this by proposing a staged probabilistic regressor (SPORE) which consists of multiple expert regressors, each one learning a subset of variations from the dataset. 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. Unlike Boosting, we marginalize the posterior prediction probabilities from each expert regressor by learning a marginalization weights regressor, where the weights are extracted during training using a Kullback-Leibler divergence-based optimization. We extend and evaluate our proposed framework for inferring hand orientation and pose simultaneously. In comparison to the state-of-the-art of hand orientation inference, multi-layered Random Forest marginalization and Boosting, our proposed method proves to be more accurate. Moreover, experimental results reveal that simultaneously learning hand orientation and pose from 2D monocular images significantly improves the pose classification performance.

Keywords: Hand Orientation; Regression; Probabilistic; Hand Pose

1. Introduction

Over recent years, real-time depth cameras have facilitated the introduction of a range of novel natural interaction methods [1, 2]. Depth maps from such cameras have been widely used in research that solves hand pose estimation under challenging settings [3, 4, 5, 6]. While depth cameras are proving 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 [7]. 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 [7]. In contrast, 2D monocular cameras are readily available in the majority of the mobile devices. Therefore, methods that utilize 2D monocular images to infer characteristics of the hand, such as hand orientation and pose, in new ways can significantly contribute towards novel interaction on these devices.

The human hand is an effective interaction tool due to its dexterous functionality in communication and manipulation [8]. For this reason, the problem of estimating hand pose has attracted a lot of research interest [3, 4, 9, 10]. Despite the recent progress in this field, limited attention has been given to study the effects of hand orientation variations on hand pose inference [1]. In this paper, we propose a method for inferring hand orientation for planar hand poses using 2D monocular images of the hand. Furthermore, we show that simultaneously learning from hand orientation and pose significantly improves the pose classification performance. We note that the proposed hand orientation inference

[1] `https://get.google.com/tango/`

[2] `http://www.pelicanimaging.com/`

method can benefit the existing model-based hand pose estimation methods that optimize against global hand orientation and pose [11, 12]. Furthermore, when used in Augmented Reality applications, the inferred hand orientation can provide the user direct control of the orientation of augmented objects [13].

We observe that the changing orientation of the hand induces changes in the projected hand shape in 2D monocular images. 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 [13]. Similar features have been previously used for hand shape-based gesture recognition [14] and person recognition [15]. As we will show in this paper, 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 than intensity and compactly encodes (as a 1D signal) the hand's global orientation unlike local feature descriptors like texture, shape context, or SIFT [16]. In such cases, a model that learns the relationship between contour-based features and the orientation angles would contribute towards understanding and using different hand postures. Furthermore, the projected hand shape is affected by a number of variations, which include inter-person hand shape and size variations, intra-person pose and style variations and self-occlusion due to varying hand orientation.

In this paper, we present a staged probabilistic regressor (SPORE) which consists of an ensemble of expert regressors, each one learning a subset of variations from the dataset. We use SPORE to address the inference of hand orientation angles, resulting from flexion/extension of the wrist and pronation/supination of the forearm measured along the azimuth and elevation axes (as shown in Fig. 1). SPORE learns the mapping of contour-based features, extracted from 2D monocular images, onto the corresponding hand orientation angles. The expert regressors in SPORE are trained, using contour-based features extracted from 2D monocular images of the hand, and added to the ensemble in stages forming an intermediate model. Evaluation of the intermediate model, using training samples, reveals a latent variable space. This

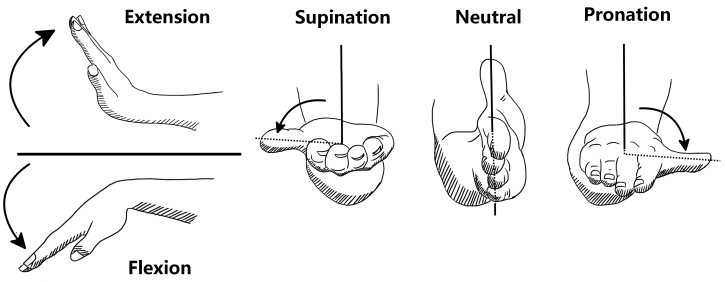


Figure 1: Movements in the wrist and forearm used to define hand orientation shows flexion and extension of the wrist and supination and pronation of the forearm.

latent variable space defines a subset of training data that the existing regressors have difficulty in learning from. This subset is used to train and add the next expert regressor. Each expert regressor gives a posterior probability for assigning a given latent variable to the training samples. These posterior probabilities are used along with the ground truth (GT) prior probability to estimate marginalization weights, which are used in the intermediate model to combine the ensemble of expert regressors. After training all stages, a marginalization weights regressor is trained that learns the mapping of hand contour-based features onto marginalization weights. Given an input silhouette image, we first extract a hand contour-based feature vector. This is followed by online prediction which involves using the feature vector to infer the marginalization weights for marginalizing the predicted posterior probabilities from each expert regressor.

1.1. Contributions

Our main contribution comes from the staged probabilistic learning, where we let the intermediate model define the subsets of data used for training the next stage. This has a two-fold contribution to the existing work in [17] where pre-defined latent variables were used for defining the subsets of the data. First, it uses the relationship of difficult to understand latent variables for defining

the subset, enabling its application to potentially any machine learning problem where easily defined subsets of the training data do not exist. Secondly, in cases where datasets are small and dividing them into subsets can result in shallow under fitting regressors, our proposed staged learning method is capable of defining latent variables with overlapping boundaries ensuring complete training of expert regressors. We further extend and demonstrate the applicability of the proposed method for simultaneously inferring hand orientation and pose. Furthermore, we are the first to show that a method which simultaneously learns hand orientation and pose from 2D images outperforms a pose only classifier as it is able to better reason the variations in pose induced due to the viewpoint changes.

The outline of this paper is as follows. Section 2 presents the related work, while Section 3 details the problem definition and Section 4 outlines the assumptions undertaken. Our proposed staged probabilistic regressor is presented in Section 5 and the experimental results with discussion are presented in Section 6. Finally, Section 7 concludes the paper.

2. Related Work

This section presents a review of the previous methods involving hand orientation and pose estimation. We include the review of hand pose estimation methods as these could be related to single-shot hand orientation estimation, where some of these methods also exploit the quantized orientation of the hand [4]. However, accurate hand orientation estimation is addressed only by a few methods [13, 18, 19]. To achieve their goals, researchers have employed different modes of input data, including colored gloves, color and depth images [8]. Our proposed SPORE method falls in the category of RGB images as we utilize colored images of hands along with the corresponding orientation angles for both training and prediction. The following sections present a brief overview of generative, discriminative and hybrid hand pose estimation methods. This is followed by the presentation of existing work on hand orientation inference. We

then present the related methods that utilize marginalization of multi-layered Random Forest (ML-RF).

2.1. Generative Methods

Generative methods use a model-based approach to address the problem of hand pose estimation. By optimizing the parameters of a hand model to the input hand image, these methods can simultaneously estimate the articulated hand orientation and pose. A major limitation of 2D monocular cameras is that the projected 2D image loses vital depth information, which gives rise to an ambiguity where it becomes difficult to differentiate multiple postures with similar 2D image projections. Generative methods are capable of addressing this ambiguity in a 2D image by utilizing a fully articulated 3D hand model [11, 12]. de La Gorce et al. [12] optimized the texture, illumination and articulations of a 3D hand model to estimate hand orientation and pose from an input 2D hand image. A similar method was proposed in [11], where generative models for both the hand and the background pixels were jointly used for image segmentation and hand pose estimation. Some of the recent generative methods also utilized depth images and advanced optimization techniques such as particle swarm optimization (PSO) [5, 20, 21]. The multi-camera based generative method in [20] recovered hand postures in the presence of occlusion from interaction with physical objects. Although these generative techniques are capable of estimating the underlying articulations corresponding to each hand posture, they are affected by the drifting problem [5, 11, 12, 20]. As the performance depends on pose estimation from previous frames, predicted poses may drift away from GT when error accumulates over time [4]. Furthermore, such methods rely on initialization, where an initial static hand orientation and pose is used. Moreover, optimizing the parameters with up to 27 degrees of freedom (DOF) for 3D hand models is computationally expensive because of the vast search space [8], and in some cases requires implementation on a GPU to achieve close to real-time execution [5]. These methods can benefit from a single-shot hand orientation and pose estimation method that can be used for initialization as well as to cor-

rect the drift in error. We note that some recent hybrid approaches described in Section 2.3 address the drifting error by re-initializing the generative approach using single-shot hand orientation and pose estimation.

2.2. Discriminative Methods

These methods are based on learning techniques and are able to learn the mapping from the feature space to target parameter space. Their ability to infer a given parameter from a single input image [22] has been a major factor in their recent popularity. Furthermore, these methods are computationally lightweight as compared to generative approaches [23].

A number of discriminative methods have been previously proposed to estimate hand pose [3, 4, 24, 25]. Wang et al. [24] used nearest neighbor search to infer hand pose from 2D monocular images. The approach relied on a colored glove and a large synthetic dataset of hand poses. In [25], a Random Forest classifier was trained on a large dataset of labeled synthetic depth images to estimate the hand pose. Keskin et al. [3] showed that the performance of the method in [25] can be improved by dividing the dataset into clusters and using the ML-RF classification. Tang et al. [26] 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. Sun et al. [10] also exploited the hierarchical relationship between different parts of the hand to train a cascaded regressor. They argued that the hand shape 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. A major challenge faced by methods relying on synthetic datasets are their lack of generalization for unseen data. Tang et al. [4] addressed this issue by proposing a semi-supervised transductive Regression Forest for articulated hand pose estimation. This approach learned hand pose from a combination of the synthetic and realistic datasets of

depth images. In [22], generalization for human body pose was addressed by incorporating real scenario-based variations into the synthetic data generation method.

Recent interest in Convolutional Neural Networks (CNN) has also been expressed in some discriminative hand pose estimation methods [27, 28, 29]. Tompson et al. [27] localized joints using CNN. They generated single-view heatmaps for joints localization using depth images as input. Ge et al. [28] extended [27] 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 CNNs 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. [29] 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.

Apart from [4], most existing discriminative hand pose estimation methods do not utilize hand orientation information. As we will show in this paper, hand orientations provide important information about variations induced in the projected 2D hand pose image due to viewpoint changes and can contribute towards improving the performance of hand pose classification.

2.3. Hybrid Methods

Recent literature has seen interest in utilizing a hybrid approach, that combines generative and discriminative methods [6, 9, 21, 27, 30]. These methods utilize the one-shot pose estimation capability of discriminative models to make generative models robust to tracking failures and drifting error. Moreover, the generative method imposes kinematic constraints resulting in realistically accurate descriptions of an articulated hand pose.

Xu et al. [31] 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 of the input data, where depth-based difference features were utilized to infer a number of candidate postures of hand. These candidate postures were used in a generative model 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. [27] used a CNN for feature extraction and to infer heatmaps for localizing joints. The heatmaps were used along with inverse kinematics 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. [9] proposed a data-driven approach to estimate 3D hand poses from depth images. This method utilized a CNN for estimating the 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. [21] utilized a discriminative re-initializer for optimizing PSO. A similar approach was proposed in [6] 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 mobile devices.

2.4. Orientation Estimation

A limited number of methods exist in the literature that estimate hand orientation [13, 18, 19]. 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 and camera setup.

To the best of our knowledge, image-based hand orientation regression has only been applied in our previous work in [13, 17], which does not require camera

calibration. Our method in [13] utilized two single-variate Random Forest (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 extended the hand orientation inference framework further, in [17], by utilizing an ML-RF regression method that used multi-variate regressors to regress the orientation angles together. Additionally, we used a hand orientation dataset that covered a more detailed orientation space. Similar to our previous work, the method proposed in this paper also does not require camera calibration which renders it suitable for a wider array of applications across different devices. The dataset used for training the proposed method comes from multiple people, which enables it to naturally handle person-to-person hand variations. The proposed staged probabilistic regression method learns different variations in stages, where it relies on intermediate model evaluations to reveal harder to learn samples.

Independent work proposed in [21] utilized global hand orientations from depth images to improve hand pose optimization. This method first generated a dataset of synthetic depth images and the corresponding global hand orientations. An ML-RF model was then utilized, where the first layer inferred a quantized hand orientation and the second layer estimated refined orientation along with additional pose information. The prediction probabilities, however, were utilized to sample candidate solutions for use with PSO-based optimization. The synthetic depth images provided detailed visible shape information, which introduced fewer ambiguities in the data as compared to 2D images, thus resulting in a simpler orientation estimation problem in [21].

2.5. Marginalization of multi-layered Random Forest

Previous work on hand pose estimation have utilized ML-RF, where complex problems have been divided and solved by a number of expert regressors trained on simpler subsets of the data [3, 7, 17]. Keskin et al. [3] proposed an ML-RF classification for hand pose estimation, which was divided into two classification

layers, namely, shape classification and pose estimation layer. Three most significant posterior probabilities from the first layer were used to marginalize the posterior probabilities in the second layer. A similar ML-RF regression method was proposed in [7], where the first layer performed coarse classification and the second layer achieved fine regression. Marginalization in this method was done using posterior probabilities from coarse classification layers as weights for predictions at the fine regression layer. Dantone et al. [32] proposed Conditional Random Forest for detecting facial features. This method also used all posterior probabilities from both layers for marginalization. Sun et al. [33] 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. All these methods relied on posterior probabilities from the first layer which tends to underestimate the true posteriors, making these methods prone to errors [34]. Furthermore, as the first layer is trained independently to the second layer, these methods cannot recover from inaccuracies arising from the posterior probabilities of the second layer. Our previous work in [17] proposed a method for learning marginalization through regression by extracting marginalization weights using posterior probabilities of the expert regressors. In this paper, we extend this work by introducing a staged probabilistic regression method for learning hand orientation.

Boosting algorithms, such as Adaboost [35] and Gradient Boosting [36], sequentially learn and combine weak learners, such as Decision Stumps, to build an expressive model. The key idea in these methods is to highlight the training samples with large errors and let the next weak learner minimize such errors. Adaboost achieves this by having an additional weight for each training sample whereas Gradient Boosting utilizes the gradient representing the global loss. Similar to Gradient Boosting, Alternating Regression Forest [37] incorporates a global loss function for improving the Regression Forest optimization algorithm. Our proposed staged learning method is inspired by Boosting, however, it differs

from Boosting as it follows a probabilistic approach. Moreover, our method utilizes only harder samples to train the subsequent stages, in contrast to all data used in non-cascaded Adaboost or Gradient Boosting. This enables our method to learn an ensemble of expert regressors, where each regressor learns well from only a subset of variations in the dataset. Furthermore, we mathematically formulate a probabilistic method for combining such ensembles, facilitating them to work collectively for better accuracy. Another appealing property of our method is that, unlike Adaboost, it does not require the underlying regressors to incorporate training weights representing the evaluation of the previously learned stages. In this paper we utilize the Random Forest as the probabilistic regressor, however, we note that our method can be easily generalized to work with any probabilistic regressor or classifier.

3. Problem Formulation

Let $\mathcal{U} = \{(\mathbf{d}_k, \mathbf{o}_k)\}_{k=1}^{K}$ be a dataset with K Contour Distance Feature (CDF) vectors $\mathbf{d}_k$ and the corresponding target orientation vectors $\mathbf{o}_k$ containing the continuous variables for azimuth (ϕ_k) and the elevation (ψ_k) angles. The CDF vectors are extracted from hand silhouette images captured from an uncalibrated 2D monocular camera such that it contains variations in hand orientation, shape and size [13]. We further describe the method for extracting the CDF in Section 5.1. In this work, we address the problem of learning the mapping of the CDF in $\mathbf{d}_k$ onto the target orientation $\mathbf{o}_k$, i.e. the orientation angle pair (ϕ_k, ψ_k). This is an ill-posed problem, as there may be multiple hand orientations that produce the same contour. We propose a staged learning algorithm for an ML-RF regressor. This method utilizes an ensemble of expert regressors that learns the complex mapping of CDF $\mathbf{d}_k$ onto the target hand orientation $\mathbf{o}_k$, despite the presence of a number of variations in orientation, shape and size of the hand.

Figure 2: Variations in style, shape and size of hand from 15 participants in our datasets. The hand images are shown for the same orientation.

4. 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 [7]. Our proposed SPORE method is 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 [3, 4, 5]. 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 method for hand orientation and pose estimation assumes the use of 2D monocular cameras, where limited computational resources are available and real-time performance is required. Moreover, to enable a method that works across different devices without the need for camera calibration, we assume that the utilized cameras are uncalibrated.

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 with a predefined set of planar hand shapes. This paper 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. In Fig. 2, we show the inter-person hand variations in style, shape and size of 15 different hands from our dataset with the same hand orientation. We note that these variations further make the hand orientation and pose estimation a challenging task.

Given the 3D normal vector, we extract the orientation encoded by azimuth (ϕ) and elevation (ψ) angles [13]. Our aim is to model variations in orientations for fronto-parallel hand, therefore we limit the orientation angles to $\sqrt{\phi^2 + \psi^2} \leq 45^\circ$. On the contrary, hand orientations with $\sqrt{\phi^2 + \psi^2} \gg 45^\circ$ are affected by self-occlusion where the visible shape of the hand is significantly occluded. Fig. 3 shows some example hand images for these orientations.

Skin and hand segmentation have a long history in computer vision, where many segmentation techniques have been devised [38, 39, 40]. We therefore extract hand silhouette images by utilizing the skin detection method proposed in [39]. We assume that the background is uncluttered and the illumination conditions are fixed for reliable silhouette extraction. This is a potential limitation of the proposed method, however, it enables us to focus on the hand orientation estimation problem given a segmented silhouette image of planar hand shape.

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 $\theta = 90^\circ$. Our assumption is satisfied by setting the operating range on the in-plane orientation to be $0^\circ < \theta < 180^\circ$.

Figure 3: 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 paper.

5. Staged Probabilistic Regression

In our proposed method, we utilize a multi-layered Random Forest 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 adds the expert regressors to the model incrementally. The flowchart of the training and prediction framework for SPORE is presented in Fig. 4. Algorithm 1 and 2 detail the training and prediction algorithm for SPORE. In the proposed framework each expert regressor that is added to the model is trained on 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. This approach enables us to use our regression-based marginalization framework without defining subsets using latent variable boundaries as in [17]. When all expert regressors have been trained, the posterior probabilities corre-

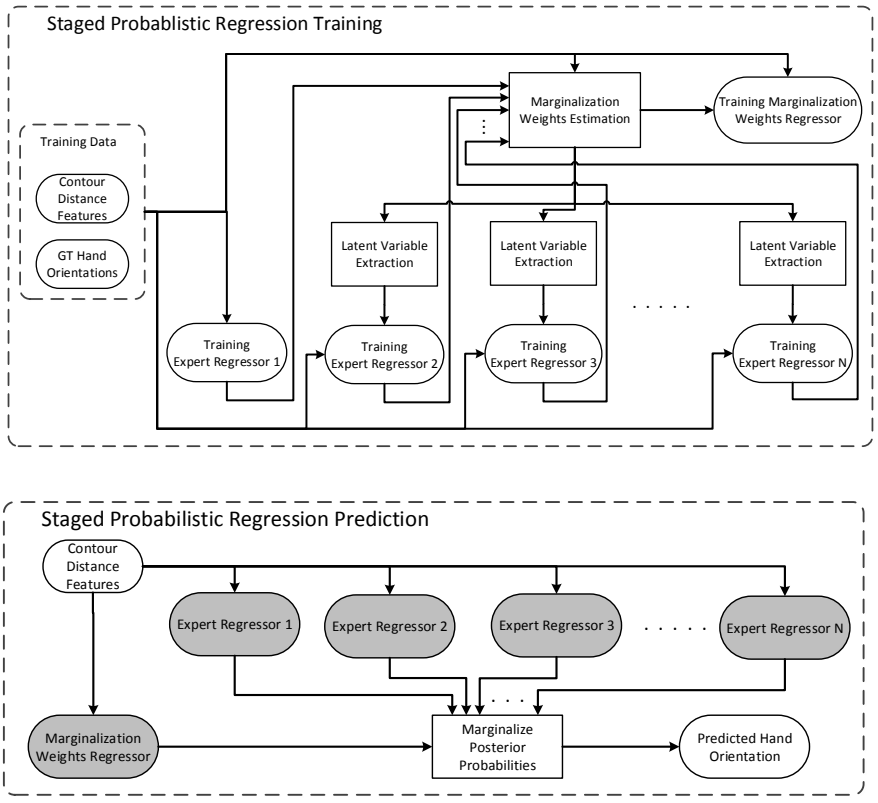


Figure 4: Flowchart shows the staged probabilistic regression (SPORE) training and prediction framework.

sponding to each sample in the training set are acquired from each of the trained expert regressors. We derive and apply a Kullback-Leibler divergence-based optimization technique 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 regressor. As demonstrated in Section 6, 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.

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, α

% N is the number of stages

% α is the error threshold

Output: ($\mathbf{ER}_n$, $\mathbf{MR}$)

% $\mathbf{ER}_n$ are N Expert Regressors

% $\mathbf{MR}$ 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 5.5

    $p(\mathbf{o}_k|\mathbf{d}_k) \leftarrow \sum_{m=1}^{n} p(\mathbf{o}_k|r_m, \mathbf{d}_k)\,\omega_{mk}$ % Marginalize probabilities described in Section 5.3

    $\mathbf{o}_p(k) \leftarrow \arg\max_{\mathbf{o}_k} p(\mathbf{o}_k|\mathbf{d}_k)$

  end

  % Define latent variable for next stage described in Section 5.4

  if $|\mathbf{o}_p(k) - \mathbf{o}_k| > \alpha$ **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 $\mathbf{MR}$

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 $\mathbf{MR}$

  $\mathbf{MR} \leftarrow$ Train($\mathcal{W}_{all}$)

end

return $\mathbf{ER}_n, \mathbf{MR}$

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

$\mathbf{o} \leftarrow \emptyset$

$\omega_n \leftarrow$ 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 $\mathbf{o}$

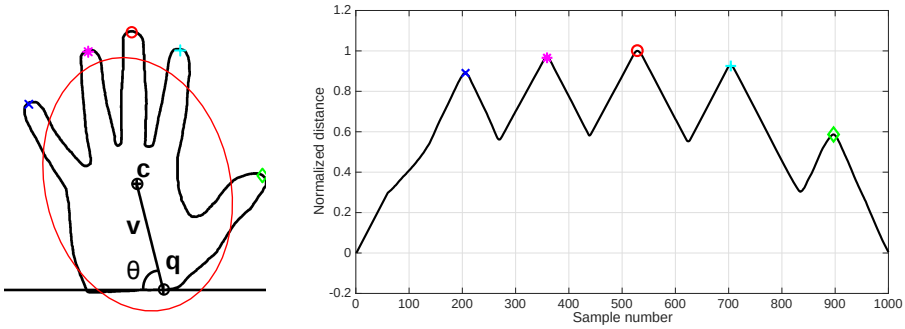
5.1. Contour Distance Features

Our proposed framework utilizes the Contour Distance Features (CDFs) which are extracted from hand silhouette images. CDFs have been previously used for hand shape-based gesture recognition [15]. The 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. We now describe the method for extracting CDF vectors.

Given a dataset $\{\mathbf{s}_k\}_{k=1}^{K}$ of input silhouette images, we compute a corresponding CDF set $\{\mathbf{d}_k\}_{k=1}^{K}$ [13]. The contour extracted from each silhouette image in $\{\mathbf{s}_k\}_{k=1}^{K}$ consists of points $\mathbf{p}_k = \{\mathbf{p}_{k1}, \cdots, \mathbf{p}_{ki}, \cdots \mathbf{p}_{kI_k}\}$, where k specifies the sample index, i is the index for each point in the contour and I_k is the total number of contour points in k^{th} sample. Let a contour distance for a single silhouette image be denoted by $\widetilde{\mathbf{d}}_k = \left\{\widetilde{d}_{k1}, \cdots, \widetilde{d}_{ki}, \cdots \widetilde{d}_{kI_k}\right\}$. $\widetilde{d}_{ki}$ 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 $\mathbf{q}_k = \{q_k^x, q_k^y\}$ and is given by:

$$\widetilde{d}_{ki} = \sqrt{(q_k^x - p_{ki}^x)^2 + (q_k^y - p_{ki}^y)^2}, \tag{1}$$

where $\mathbf{q}_k$ is extracted, for each sample in $\{\mathbf{s}_k\}_{k=1}^{K}$, by emanating a ray from



(a) (b)

Figure 5: Contour Distance Feature (CDF) vector extraction from a hand contour showing (a) the method for extraction of a 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.

centroid in the direction of the wrist [13]. We further discuss the approach for extracting $\mathbf{q}_k$ in the next section. The extracted features have a different number of samples I_k and magnitude depending on the scale changes and inter-person hand shape variations. We normalize the magnitude using Equation 2.

$$\overline{\mathbf{d}}_k = \frac{\widetilde{\mathbf{d}}_k}{\max\limits_{1 \leq i \leq I_k} (\widetilde{d}_{ki})}. \tag{2}$$

$\overline{\mathbf{d}}_k$ 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 value of I_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.

5.1.1. 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 1 to extract the CDF vector. Furthermore, the point $\mathbf{q}_k$ also aligns the corresponding CDF vector. Fig. 5 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. Given θ and the contour centroid $\mathbf{c}_k$, an equation of a ray emanating from $\mathbf{c}_k$ can be defined by:

$$\mathbf{v}_k = \xi\kappa\hat{\mathbf{v}}_k + \mathbf{c}_k, \tag{3}$$

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

$$\hat{\mathbf{v}}_k = \frac{\begin{bmatrix} 1 \\ \tan\theta \end{bmatrix}}{\sqrt{1^2 + \tan^2\theta}}, \tag{4}$$

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

$$\xi = \begin{cases} +1 & \text{if } \theta < 90^\circ \\ -1 & \text{if } \theta \geq 90^\circ, \end{cases} \tag{5}$$

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

The direction scalar ξ is calculated using Equation 5 based on the assumption that the in-plane orientation θ of the hand will always be in the range $0^\circ < \theta < 180^\circ$. ξ is used in Equation 3 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. The construction of CDF in this way makes the proposed method invariant to in-plane rotations in the range $0^\circ < \theta < 180^\circ$.

5.2. Random Forest Construction

Building on the reported superior performance in the existing work for hand pose estimation [3, 4, 7], our proposed staged probabilistic regression method utilizes a Random Forest training algorithm for both regression layers. In this section, we present details of the training algorithm specific to our proposed method, a further in-depth literature on Random Forest can be found in [41].

The forest is a collection of T trees which are trained using a training dataset $\mathcal{U} = \{(\mathbf{d}_k, \mathbf{o}_k)\}_{k=1}^{K}$. Each tree consists of split nodes, 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. 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 split using a splitting function $f(\mathcal{U}_j, \Theta)$ defined as:

$$f(\mathcal{U}_j, \Theta) = \begin{cases} Left & \text{if } \mathcal{U}_j(w) < \tau, \\ Right & \text{otherwise.} \end{cases} \tag{6}$$

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

$$Q(\mathcal{U}_j, \Theta) = H(\mathcal{U}_j) - \sum_{b \in \{Left, Right\}} \frac{|\mathcal{U}_j^b|}{|\mathcal{U}_j|} H\left(\mathcal{U}_j^b\right), \tag{7}$$

where $H(\mathcal{U}_j)$ is the Shannon entropy of $\mathcal{U}_j$.

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, a given input feature vector $\mathbf{d}$ propagates down the branches of each tree, where a leaf node gives a posterior probability $p_t(\phi, \psi|\mathbf{d})$. The predictions from all trees are aggregated as:

$$p(\phi, \psi|\mathbf{d}) = \frac{1}{T} \sum_{t=1}^{T} p_t(\phi, \psi|\mathbf{d}), \tag{8}$$

where (ϕ, ψ) is the orientation vector $\mathbf{o}$ whose final value is determined by maximum-a-posteriori (MAP) estimation as:

$$(\phi, \psi)^* = \arg\max_{\phi, \psi} p(\phi, \psi|\mathbf{d}). \tag{9}$$

5.3. Marginalization of Multiple Expert Regressors

In our proposed method, the ensemble of expert regressors consists of a set of multi-variate Random Forest regressors that are trained on the subset of our

hand orientation dataset $\mathcal{U}$. This ensemble of expert regressors enables better generalization in the presence of a number of variations in the dataset. The subsets of our dataset are defined based on latent variable representations that are generated using the intermediate model evaluations. Given an input CDF vector $\mathbf{d}$ each expert regressor infers the posterior probability $p(\phi, \psi | r_n, \mathbf{d})$ for a given latent variable r_n.

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

$$p(\phi, \psi | \mathbf{d}) = \sum_{n=1}^{N} p(\phi, \psi | r_n, \mathbf{d}) \, \omega_n, \tag{10}$$

where ω_n are marginalization weights corresponding to each latent variable such that $\sum_{n=1}^{N} \omega_n = 1$ and N is the total number of expert regressors. In the subsequent sections, we present a method to estimate the marginalization weights ω_n from trained expert models and propose to use a marginalization weights regressor that learns the mapping of CDF $\mathbf{d}$ onto the corresponding marginalization weights ω_n.

5.4. Latent Variable Generation using Intermediate Models

In our proposed work we do not explicitly define the latent variable space, as in [17]. We, however, rely on intermediate model evaluations for defining a latent variable r_n and, as a result, define the subsets used for training the expert regressor in the n^{th} stage. We start training the first expert regressor using all samples in the dataset $\mathcal{U}$. 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 $\mathcal{U}$, we determine if it belongs to the latent variable r_n by:

$$r_n(k) = \begin{cases} 1 & \text{if } |\mathbf{o}_p(k) - \mathbf{o}_k| > \alpha, \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

where $\mathbf{o}_p(k)$ are the orientation angles predicted by marginalizing intermediate model probabilities using Equation 10 and $\mathbf{o}_k$ are the GT orientation angles, respectively. α 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 [17]. Firstly, 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. Secondly, in cases where datasets are small and dividing them into subsets can result in shallow under fitting models, our proposed incremental learning method is capable of defining latent variables with overlapping boundaries ensuring complete training of expert regressors.

5.5. Marginalization through Regression

We marginalize the posterior probabilities from multiple expert regressors using a single Random Forest regressor. This regressor is trained using marginalization weights that are extracted using training data. Marginalization through regression is able to generalize better by learning a complex mapping of the CDF vectors onto weights that marginalize the posterior probabilities from expert regressors [17]. For estimating the marginalization weights, we first formulate the prior probability for the training samples using the GT orientation angles (ϕ_{gt}, ψ_{gt}) in a multi-variate normal distribution as:

$$p\left(\phi_{gt}, \psi_{gt}\right) = \mathcal{N}\left(\left(\phi_{gt}, \psi_{gt}\right), \Sigma\right), \tag{12}$$

where Σ is the covariance that can be adjusted to control the spread of $p\left(\phi_{gt}, \psi_{gt}\right)$.

Given the prior probability $p\left(\phi_{gt}, \psi_{gt}\right)$ and the corresponding posterior probabilities $p\left(\phi, \psi | r_n, \mathbf{d}\right)$, we propose a novel optimization method, where the marginalization error is based on the Kullback-Leibler divergence [42]. Fig. 6 shows the marginalization weights estimation framework. The error is optimized to estimate the GT marginalization weights ω_n for all latent variables $r_n \in \{r_1, r_2, r_3 \cdots r_N\}$.

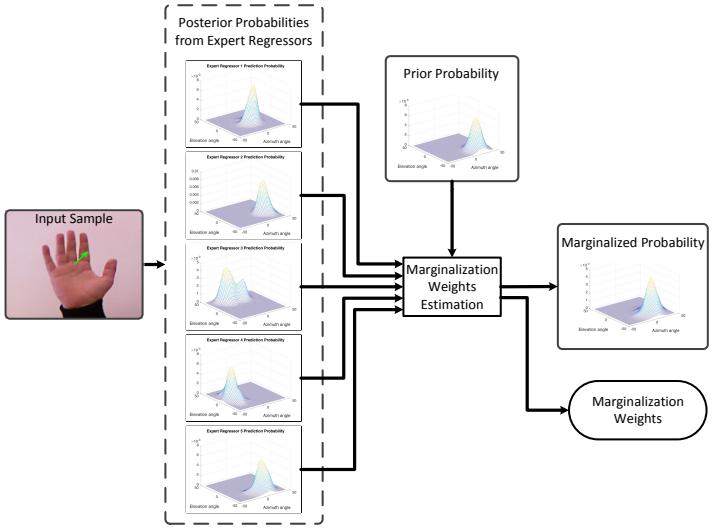


Figure 6: Marginalization weights estimation using training data. A training sample is used to get posterior probabilities from each expert regressor. These probabilities are then used along with the prior probability in Equation 13 to estimate marginalization weights and the corresponding marginalized probability. Probabilities shown are only for demonstrating the concept and are not actual probabilities from multiple stages of SPORE.

We define this error as:

$$E = \iint p\left(\phi_{gt}, \psi_{gt}\right) \log \frac{p\left(\phi_{gt}, \psi_{gt}\right)}{p\left(\phi, \psi | \mathbf{d}\right)} d\phi d\psi. \tag{13}$$

Derivation We optimize the weights using gradient descent, which relies on derivatives of E with respect to the weights ω_n. Here we present the derivation of partial derivatives from Equation 13 that can be used to obtain optimal weights ω_n.

$$E = \iint p\left(\phi_{gt}, \psi_{gt}\right) \log \frac{p\left(\phi_{gt}, \psi_{gt}\right)}{p\left(\phi, \psi | \mathbf{d}\right)} d\phi d\psi, \tag{14}$$

$$= \iint p\left(\phi_{gt}, \psi_{gt}\right) [\log\ p\left(\phi_{gt}, \psi_{gt}\right) - \log(\sum_{n=1}^{N} p\left(\phi, \psi | r_n, \mathbf{d}\right) \omega_n)] d\phi d\psi. \tag{15}$$

The partial derivative w.r.t ω_n can then be defined as:

$$\frac{\partial E}{\partial \omega_n} = -\iint \frac{p\left(\phi_{gt}, \psi_{gt}\right) p\left(\phi, \psi | r_n, \mathbf{d}\right)}{\sum_{n=1}^{N} p\left(\phi, \psi | r_n, \mathbf{d}\right) \omega_n} d\phi d\psi. \tag{16}$$

Optimization We use gradient descent with:

$$\nabla E = \left[\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \frac{\partial E}{\partial w_3} \cdots \frac{\partial E}{\partial w_N}\right], \tag{17}$$

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

$$\omega_n^{\gamma+1} = \omega_n^{\gamma} - \lambda \nabla E^{\gamma}, \tag{18}$$

where λ is the step size along the negative gradient direction and γ is the iteration number. At this stage, we have the optimal weights fit to the GT. These are required to train the marginalization weights regressor that produces the weights ω_n during online prediction. This regressor is described next.

Marginalization weights regressor We use a multi-variate Random Forest regressor to learn the mapping of CDF vectors to marginalization weights ω_n. This regressor is used during prediction to infer marginalization weights ω_n for marginalizing the posterior probabilities $p\left(\phi, \psi | r_n, \mathbf{d}\right)$ from each expert regressors using Equation 10.

5.6. Extension to Estimate Orientation and Pose

The proposed staged probabilistic regression method can be extended to simultaneously infer the hand orientation and pose. To achieve this, we utilize a hand orientation and pose dataset which contains the CDF ($\mathbf{d}_k$), the corresponding 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 $p\left(\chi | \mathbf{d}\right)$ in the leaf nodes. Training of this extended model is driven by both orientation regression as well as pose classification data. We achieve this by using a selected information gain Q_s, which is determined by:

$$Q_s = (1 - \beta) Q_r + \beta Q_c, \tag{19}$$

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 [41].

Given the additional pose classification task, we define the latent variable space r_n by modifying Equation 11 with an additional term as:

$$r_n(k) = \begin{cases} 1 & \text{if } |\mathbf{o}_p(k) - \mathbf{o}_k| > \alpha \text{ or } \chi_p(k) \neq \chi_k, \\ 0 & \text{otherwise,} \end{cases} \tag{20}$$

where $\chi_p(k)$ and χ_k are the predicted and GT hand poses, respectively. The additional criteria related to hand poses in Equation 20 identifies samples for which the existing intermediate model has difficulty in inferring the hand pose.

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

$$p(\chi|\mathbf{d}) = \sum_a p(\chi|r_n, \mathbf{d})\, \rho_n, \tag{21}$$

where ρ_n are weights corresponding to each latent variable for the classification posterior probabilities and $\sum_n^N \rho_n = 1$. We estimate these marginalization weights using discrete version of energy E defined as:

$$E_c = \sum_{\chi} p(\chi_{gt}) \log \frac{p(\chi_{gt})}{p(\chi|\mathbf{d})}. \tag{22}$$

The partial derivatives w.r.t ρ_n can be defined using E_c as:

$$\frac{\partial E_c}{\partial \rho_n} = -\sum_{\chi} \frac{p(\chi_{gt})\, p(\chi|r_n, \mathbf{d})}{\sum_{n=1}^N p(\chi|r_n, \mathbf{d})\, \rho_n}. \tag{23}$$

We use gradient descent to estimate the optimal weights ρ_n for the classification posterior probabilities. We augment the marginalization weights for classification ρ_n and regression ω_n to train a marginalization weights regressor that infers both weights simultaneously.

(a) χ_1 (b) χ_2 (c) χ_3 (d) χ_4

Figure 7: 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. Experimental Validation

We evaluate our proposed staged probabilistic regression (SPORE) method using two datasets collected from 22 participants. The first dataset referred to as *single pose dataset* herein, contains 9414 samples captured for an open hand pose from 22 different participants. The second dataset, referred to as *multiple pose dataset* herein, contains 8675 samples captured using four different hand poses (shown in Fig. 7) 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. All of the hand poses used in this paper are planar, which enables us to extract reliable GT hand orientation using the method described in [13]. The range of the orientation angles captured by these datasets are restricted to a circular space defined by $\sqrt{\phi^2 + \psi^2} \leq 45°$. This gives us an appropriate ratio for the number of samples against the variations within the defined orientation space. We show experimental results that demonstrate the ability of our proposed staged probabilistic regression method to infer hand orientation and pose on these datasets.

6.1. 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-SV

RF) with independence assumption on each hand orientation angle [13]. We also compare with four different methods for the marginalization of ML-RF regressors [7, 17, 32]. Furthermore, as SPORE is inspired by Boosting, we compare it with Random Forest with Adaboost (RF Adaboost) [35], Alternating Regression Forest (ARF) [37] and Gradient Boosted Trees (GBT) [36]. Our previous work proposed in [17], 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 similar to the one described in Section 5. 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 that are discussed in Section 6.4. We refer to the other ML-RF marginalization methods as ML-RF1, ML-RF2 and ML-RF3 herein, adapted from [7] and [32]. These methods also rely on the same explicit definition of latent variables as in ML-RF MtR. While the methods proposed in [7] and [32] do not originally address hand orientation regression problem, they provide a method for marginalizing the ML-RF in different domains. In our experimental validation, these three ML-RF comparison methods use a two-layered Random Forest 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, as defined by Equations 24 and 25.

$$r_n^* = \arg\max_{r_n} p\left(r_n|\mathbf{d}_k\right), \tag{24}$$

$$(\phi^*, \psi^*) = \arg\max_{(\phi,\psi)} p\left(\phi, \psi|r_n^*, \mathbf{d}_k\right). \tag{25}$$

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. The mathematical formulation for predictions using ML-RF2 is shown in Equation 26.

$$(\phi^*, \psi^*) = \sum_{n=1}^{N} p\left(r_n|\mathbf{d}_k\right) \quad \arg\max_{(\phi,\psi)} p(\phi, \psi|r_n, \mathbf{d}_k), \tag{26}$$

where $N = 5$ is the total number of expert regressors in the ML-RF model. Equations 27 and 28 show the formulation for making predictions using ML-RF3.

$$p(\phi, \psi|\mathbf{d}_k) = \sum_{n=1}^{N} p\left(r_n|\mathbf{d}_k\right) \; p(\phi, \psi|r_n, \mathbf{d}_k), \tag{27}$$

$$(\phi^*, \psi^*) = \arg\max_{(\phi,\psi)} p(\phi, \psi|\mathbf{d}_k). \tag{28}$$

We evaluate the extension of our proposed method to simultaneously estimate orientation and pose using the *multiple pose dataset*. To show the role of hand orientation in improving the pose classification performance we compare this extension of our work with a Random Forest classifier (RF Clf) that infers hand pose only. We also make the comparison of orientation inference of this extension with all of the comparison methods that utilize Random Forest. These include ML-RF MtR, SL-RF SV, ML-RF1, ML-RF2, ML-RF3, RF Adaboost and ARF. We exclude evaluation of GBT on this data as this method does not provide a way to combine regression and classification into the same model. The results of these comparisons are discussed in Section 6.5.

6.2. Error Measures

We evaluate the proposed method 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 the quantitative measures below.

6.2.1. 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 Equations 29 and 30.

$$\phi_m = \frac{\sum_{k=1}^{K} |\phi_k - \phi_{pk}|}{K}, \tag{29}$$

$$\psi_m = \frac{\sum_{k=1}^{K} |\psi_k - \psi_{pk}|}{K}. \tag{30}$$

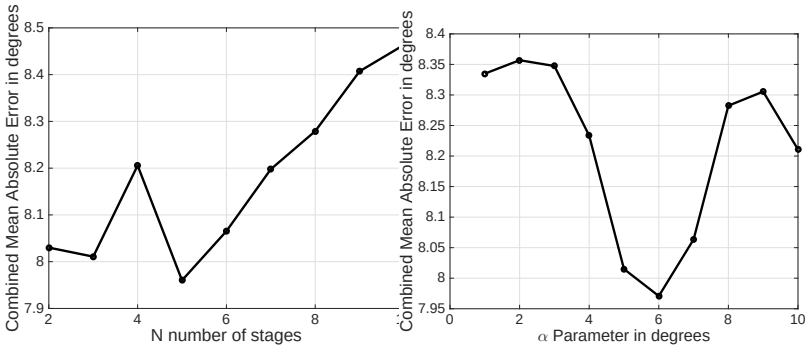
We use MAE instead of Euclidean distance between the GT and predicted orientation as in our work we found that sometimes the regressor is able to infer only one of the two angles correctly. In such a scenario, a Euclidean distance does not present accurate measure of performance. On the other hand, MAE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

$$CMAE = \frac{\phi_m + \psi_m}{2}, \tag{31}$$

CMAE is particularly used for tuning different training parameters of SPORE.

6.3. Parameter Optimization

The proposed SPORE method has different training parameters. These include the number of trees (T), depth of each tree (δ_t), minimum number of samples in each leaf node (η_j), the number of features selected at each split node (ϵ), the number of stages (N), the latent variable generation parameter α and the probability $p(\beta)$ for selecting information gain for the extension of the proposed method for simultaneous hand orientation and pose inference. As all comparison methods utilize Random Forest, therefore we empirically set the



(a) (b)

Figure 8: 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) *single pose dataset* and (b) using *multiple pose dataset*.

values of the related parameters as, $T = 100$, $\delta_t = 10$, $\eta_j = 20$, $\epsilon = 1$. As the proposed SPORE method is independent of the number of predefined subsets, therefore any number of stages N can be used. We perform single-fold validation using the *single pose dataset*, randomly selecting 70% of the data for training and 30% for testing, to evaluate the optimal values for N, α and $p(\beta)$.

The CMAE with varying number of stages N is shown in Fig. 8 (a). It can be seen that SPORE with $N = 5$ stages presents the minimum MAE for both azimuth (ϕ) and elevation (ψ) angles combined. The error increases for $N \gg 5$ as the subsequent regression stages with $N > 5$ do not get enough data for training. Hence, $N = 5$ optimally captures the variations in our dataset by providing a good balance for the number of stages and sufficient samples in the subsets defined by the corresponding latent variables. We choose $N = 5$ for the rest of the experimental validation. Fig. 8 (b) shows the CMAE with varying α threshold in Equation 11 using $N = 5$. We note that selecting $\alpha = 6^\circ$ yields the best performance of the proposed SPORE method. α acts as a threshold for defining the subset of training data for the next stage. We observe that if α is too low, i.e. $\alpha \approx 0$, then the subsequent stages will all be trained using all training samples, thus not targeting to learn from specific variations. On the contrary, if α is set too high, i.e. $\alpha > 10^\circ$, then the latent variable space will not

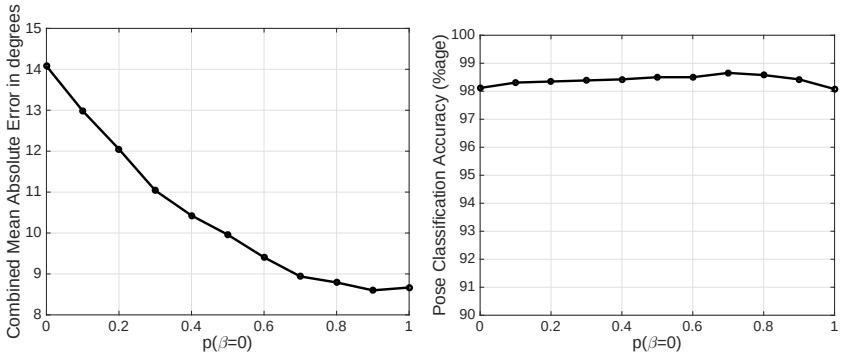


Figure 9: Parameter optimization for $p(\beta = 0)$ shows evaluation of the proposed SPORE method with hand orientation and pose estimation extension. (a) presents Combined Mean Absolute Error (CMAE) for orientation inference and (b) shows the accuracy of pose classification against varying probability $p(\beta = 0)$ of selecting classification or regression information gain.

be fully defined for subsequent stages, hence resulting in under fitting models. We note that $\alpha = 6^\circ$ maintains a good balance for selecting harder samples for training subsequent stages. Therefore we select this value for the rest of the experimental validation.

The extension of our proposed SPORE method for simultaneously inferring hand orientation and 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. 9. We note that selecting regression information gain more often than classification 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 used ($p(\beta = 0) = 1$). This is because the information for each pose is well encoded within the CDF and hand orientation. In our experimental validation we use $p(\beta = 0) = 0.9$. This means that at each split node, regression information gain is selected more frequently than classification information gain. As we will further demonstrate

Table 1: Mean Absolute Error (MAE) in degrees for single pose experimental validation in Section 6.4.

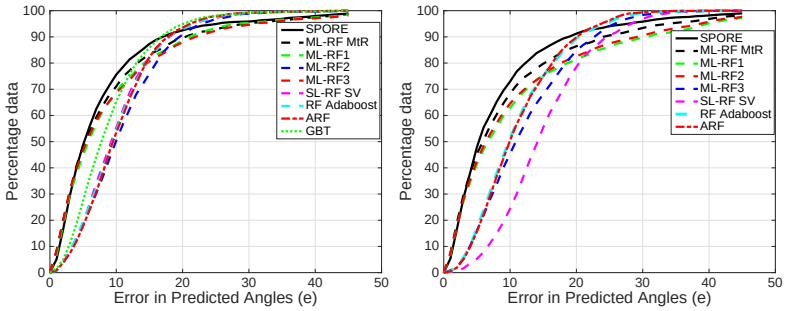
Method used	**Azimuth (ϕ)**		**Elevation (ψ)**	
		p-value		p-value
SPORE (proposed)	**8.42°**	-	**7.38°**	-
ML-RF MtR [17]	9.65°	**0.00**	7.81°	$\mathbf{0.13 \times 10^{-10}}$
SL-RF SV [13]	11.58°	$\mathbf{0.25 \times 10^{-8}}$	8.75°	**0.00**
RF Adaboost [35]	11.54°	$\mathbf{0.72 \times 10^{-10}}$	9.06°	**0.00**
ML-RF1	10.24°	$\mathbf{0.22 \times 10^{-5}}$	8.02°	**0.00**
ML-RF2	12.82°	$\mathbf{0.20 \times 10^{-3}}$	9.12°	$\mathbf{0.11 \times 10^{-2}}$
ML-RF3	10.45°	$\mathbf{0.10 \times 10^{-20}}$	8.13°	$\mathbf{0.15 \times 10^{-18}}$
ARF [37]	11.67°	$\mathbf{0.29 \times 10^{-2}}$	9.00°	**0.00**
GBT [36]	10.39°	$\mathbf{0.96 \times 10^{-3}}$	7.62°	$\mathbf{0.90 \times 10^{-4}}$

in Section 6.5, the hand orientation information can significantly improve pose classification results as with orientation the SPORE model is able to build a better understanding of the hand pose dataset.

6.4. Experimental Validation using Single Pose Dataset

The evaluation of our proposed hand orientation inference method is done using the *single pose dataset*. We perform single-fold validation by randomly dividing 70% of the data into the training set and using the remaining 30% for testing. Table 1 shows the MAE in degrees for the single-fold evaluation using the proposed SPORE method and the comparison methods. Furthermore, we also show in Fig. 10 (a) the percentage of data that lies under a given error in prediction.

We note that the proposed staged probabilistic regression outperforms the existing state-of-the-art in ML-RF marginalization as well as hand orientation inference. The proposed method also outperforms the method related to Boosting, namely, RF Adaboost, ARF and GBT. These methods lack a probabilistic



(a) (b)

Figure 10: Percentage data vs error in prediction shows the percentage of data that lies below a given error in prediction for the single-fold validation using (a) single pose dataset and (b) using multiple pose dataset.

approach resulting in higher MAE. On the contrary, the proposed method is formulated using probabilities, where the complex mapping between each stage and the input features is learned. We further notice from Fig. 10 (a) that the proposed staged probabilistic regression performs better with 78% of data lying in under 10° of error. We also note that at around 20° of error, the ML-RF2, SL-RF SV, RF Adaboost, ARF and GBT contain more percentage data than any other method. This is due to the fact that all other comparison methods, including the proposed SPORE, contains 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. This affects the regressors where for a given hand contour, the regressors infer symmetrically opposite hand orientations. This problem shows up in all methods that use a probabilistic approach for marginalization. ML-RF2, SL-RF SV, RF Adaboost, ARF and GBT 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 1. These models have a bias as they are unable to fully learn from all the variations within the orientation dataset.

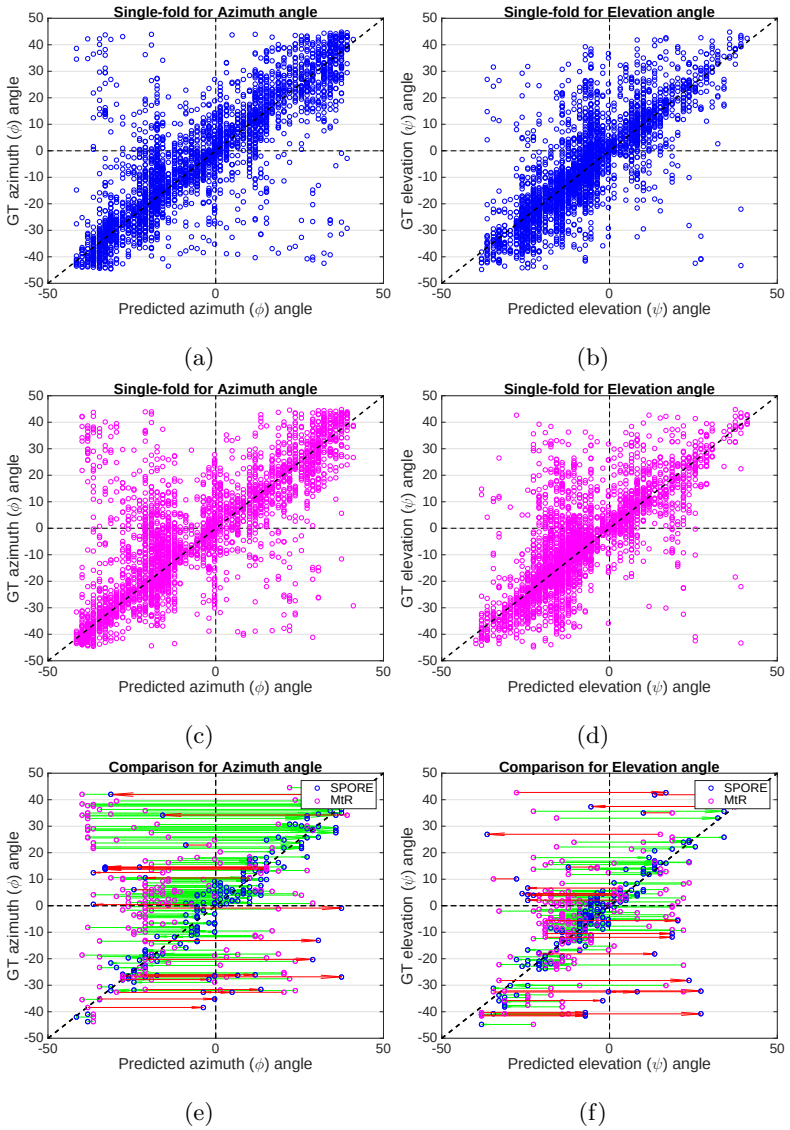


Figure 11: Ground Truth (GT) versus predicted orientation angle plots showing results for (a)-(b) the proposed SPORE method and (c)-(d) the ML-RF MtR method proposed in [17]. (e)-(f) shows the errors in ML-MtR that were corrected by SPORE (green arrows) and the correct predictions by ML-MtR that were incorrectly inferred by SPORE (red arrows). The larger number green lines compared to red show that SPORE improves estimation for the majority of samples.

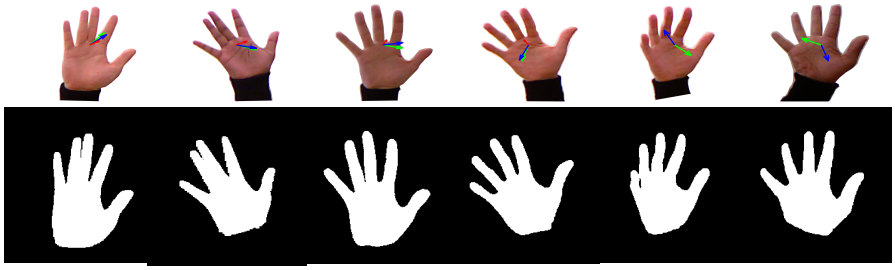


Figure 12: Success and Failure cases for the proposed SPORE method. The GT orientation (green) and predicted orientation using SPORE (blue) and ML-RF MtR (red) are shown with arrows. The first row shows the color images, whereas the corresponding silhouette images are shown in the second row. (a)-(d) shows success cases where the proposed SPORE method successfully able to infers the orientation. (e)-(f) shows the failure cases where the proposed method fails.

SPORE produces the results with the least error and a paired t-test with p-value less than 0.05 demonstrates that SPORE's improvement over all other methods is statistically significant.

We also present the comparison of the proposed SPORE method with the most closely related ML-RF MtR method proposed in [17]. In Fig. 11, we present the single-fold validation results showing the GT versus predicted plots for the proposed SPORE method and the ML-RF MtR method. Fig 11 (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. We note that in this comparison a number of 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. This 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. Fig. 12

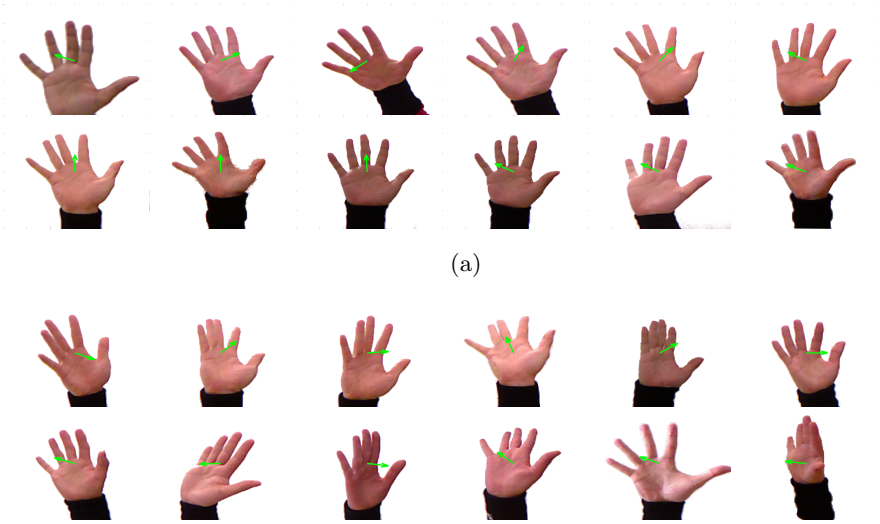


Figure 13: Easy versus hard training samples. (a) shows easy training samples that are successfully learned from in the first regressor with error $|\mathbf{o}_p(k) - \mathbf{o}_k| < \alpha$. (b) shows harder training samples, with error $|\mathbf{o}_p(k) - \mathbf{o}_k| > \alpha$, that are not completely learned from in the first expert regressor and hence are selected for the next stage training. Green arrows show the GT orientation. The difference between easy and hard samples can be seen in terms of inter-person pose, shape and style variation.

shows success and failure cases for the proposed SPORE method. We observe that the proposed method fails on difficult samples where the fingers are not completely outstretched (Fig. 12 (e)-(f)). Moreover, in Fig. 13 we present the easy versus harder to learn hand orientation samples. In Fig. 13 (a) easy samples are presented that the SPORE learns from in the first stage. Fig. 13 (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.

6.5. Experimental Validation using Multiple Pose Dataset

We use the *multiple pose dataset* to evaluate the extension of our proposed staged probabilistic regressor for inferring both hand orientation and pose simul-

Table 2: Mean Absolute Error (MAE) in degrees for multiple pose experimental validation in Section 6.5.

Method used	**Azimuth (ϕ)**		**Elevation (ψ)**	
		p-value		p-value
SPORE (proposed)	**8.53°**	-	**8.14°**	-
ML-RF MtR [17]	9.63°	$\mathbf{0.41x10^{-11}}$	9.77°	**0.00**
SL-RF SV [13]	15.04°	$\mathbf{0.33x10^{-8}}$	14.95°	$\mathbf{0.92x10^{-10}}$
RF Adaboost [35]	11.52°	$\mathbf{0.29x10^{-16}}$	10.77°	$\mathbf{0.32x10^{-13}}$
ML-RF1	11.20°	$\mathbf{0.22x10^{-5}}$	11.43°	**0.00**
ML-RF2	12.83°	$\mathbf{0.31x10^{-5}}$	11.63°	$\mathbf{0.11x10^{-6}}$
ML-RF3	11.00°	$\mathbf{0.33x10^{-16}}$	10.81°	**0.00**
ARF [37]	11.51°	$\mathbf{0.4x10^{-10}}$	10.83°	$\mathbf{0.47x10^{-13}}$

taneously. The MAE in degrees for the single-fold evaluation using this extension and the comparison methods is presented in Table 2. Fig. 10 (b) shows the percentage of data that lies under a given error in prediction for SPORE and the comparison methods. We notice that again, the proposed SPORE outperforms the comparison methods that infer hand orientation and pose simultaneously. A paired t-test with p-value less than 0.05 shows that improvement in orientation predictions using SPORE are statistically significant as compared to the comparison methods.

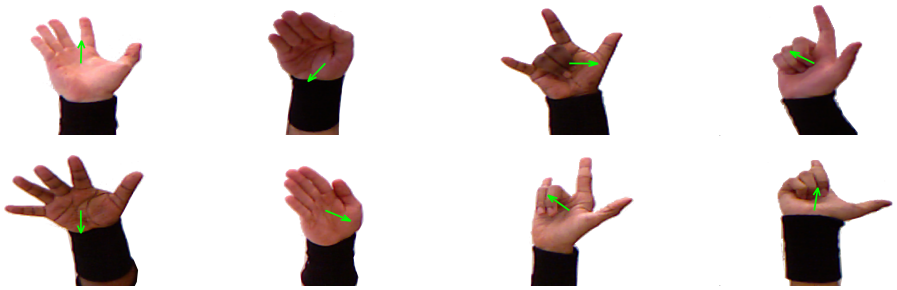
Table 3: Hand pose classification results using SPORE.

		Predicted Pose			
		χ_1	χ_2	χ_3	χ_4
GT Pose	χ_1	97.94%	0.00%	1.74%	0.32%
	χ_2	0.00%	99.66%	0.17%	0.17%
	χ_3	0.44%	0.00%	98.52%	1.03%
	χ_4	0.14%	0.56%	1.69%	97.61%

Table 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%

Furthermore, we compare the pose classification accuracy of the proposed



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

Figure 14: 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. This orientation information is not used for RF Clf training.

SPORE method with RF Clf that learns only the pose classification. We present confusion matrices for these results in Table 3 and Table 4, respectively. It can be seen that the proposed SPORE method outperforms an RF Clf for the pose classification task. This is due to the presence of the additional orientation information that the SPORE method 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. 14 we present the samples that are misclassified by RF Clf due to the absence of orientation information. These results let us understand the importance of hand orientation in hand pose classification in 2D images. We note that when such orientation information is not present, then the classifiers have difficulty in hand pose classification under varying viewpoint.

This paper focuses on using SPORE for hand orientation and pose inference. We observe that the proposed method is generalizable to other domains. SPORE can be used with any probabilistic regressor or classifier, where the dataset contains large variations that are not fully captured with a single model.

7. Conclusion

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. The 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 Contour Distance Features 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 the marginalization of multi-layered Random Forest, hand orientation inference and Boosting. Furthermore, we show that a method which simultaneously learns from hand orientation and pose outperforms pose only classification as it is able to better understand the variations in pose induced due to viewpoint changes. Our future work focuses on introducing a bigger vocabulary of hand poses, application of SPORE in other domains and the introduction of a temporal coherence method that addresses the symmetry problem. Exploring effective CNN architectures for simultaneous hand orientation and pose estimation is another interesting future direction for our work.

References

[1] J. S. Supancic, G. Rogez, Y. Yang, J. Shotton, D. Ramanan, Depth-based hand pose estimation: data, methods, and challenges, in: Proceedings of

the IEEE international conference on computer vision, 2015, pp. 1868–1876.

[2] J. Han, L. Shao, D. Xu, J. Shotton, Enhanced computer vision with microsoft kinect sensor: A review, Cybernetics, IEEE Transactions on 43 (5) (2013) pp.1318–1334.

[3] C. Keskin, F. Kıraç, Y. E. Kara, L. Akarun, Hand pose estimation and hand shape classification using multi-layered randomized decision forests, in: Computer Vision–ECCV 2012, Springer, 2012, pp. 852–863.

[4] D. Tang, T.-H. Yu, T.-K. Kim, Real-time articulated hand pose estimation using semi-supervised transductive regression forests, in: IEEE International Conference on Computer Vision, 2013, pp. 3224–3231.

[5] I. Oikonomidis, N. Kyriazis, A. A. Argyros, Efficient model-based 3d tracking of hand articulations using kinect., in: British Machine Vision Conference, Vol. 1, 2011, p. 3.

[6] J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, T. Sharp, E. Soto, D. Sweeney, J. Valentin, B. Luff, et al., Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences, ACM Transactions on Graphics (TOG) 35 (4) (2016) 143.

[7] S. R. Fanello, C. Keskin, S. Izadi, P. Kohli, D. Kim, D. Sweeney, A. Criminisi, J. Shotton, S. B. Kang, T. Paek, Learning to be a depth camera for close-range human capture and interaction, ACM Transactions on Graphics (TOG) 33 (4) (2014) 86.

[8] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, X. Twombly, Vision-based hand pose estimation: A review, Computer Vision and Image Understanding 108 (1) (2007) pp.52–73.

[9] M. Oberweger, P. Wohlhart, V. Lepetit, Training a feedback loop for hand pose estimation, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3316–3324.

[10] X. Sun, Y. Wei, S. Liang, X. Tang, J. Sun, Cascaded hand pose regression, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 824–832.

[11] M. de La Gorce, N. Paragios, A variational approach to monocular hand-
695 pose estimation, Computer Vision and Image Understanding 114 (3) (2010) pp.363–372.

[12] M. de La Gorce, D. J. Fleet, N. Paragios, Model-based 3d hand pose estimation from monocular video, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (9) (2011) 1793–1805.

700 [13] M. Asad, G. Slabaugh, Hand orientation regression using random forest for augmented reality, in: International Conference on Augmented and Virtual Reality, Springer, 2014, pp. 159–174.

[14] Z. Ren, J. Yuan, J. Meng, Z. Zhang, Robust part-based hand gesture recognition using kinect sensor, IEEE transactions on multimedia 15 (5) (2013)
705 1110–1120.

[15] E. Yoruk, E. Konukoglu, B. Sankur, J. Darbon, Shape-based hand recognition, IEEE Transactions on Image Processing 15 (7) (2006) 1803–1815.

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

710 [17] M. Asad, G. Slabaugh, Learning marginalization through regression for hand orientation inference, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2016, pp. 10–18.

[18] T. Lee, T. Höllerer, Handy ar: Markerless inspection of augmented reality objects using fingertip tracking, in: IEEE International Symposium on
715 Wearable Computers, IEEE, 2007, pp. 83–90.

[19] Y. Mizuchi, Y. Hagiwara, A. Suzuki, H. Imamura, Y. Choi, Monocular 3d palm posture estimation based on feature-points robust against finger

motion, in: International Conference on Control, Automation and Systems (ICCAS), IEEE, 2013, pp. 1014–1019.

[20] I. Oikonomidis, N. Kyriazis, A. 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), IEEE, 2011, pp. 2088–2095.

[21] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. 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, ACM, 2015, pp. 3633–3642.

[22] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, R. Moore, Real-time human pose recognition in parts from single depth images, Communications of the ACM 56 (1) (2013) 116–124.

[23] R. Rosales, S. Sclaroff, Combining generative and discriminative models in a framework for articulated pose estimation, International Journal of Computer Vision 67 (3) (2006) 251–276.

[24] R. Y. Wang, J. Popović, Real-time hand-tracking with a color glove, in: ACM Transactions on Graphics (TOG), Vol. 28, ACM, 2009, p. 63.

[25] C. Keskin, F. Kıraç, Y. E. Kara, L. Akarun, Real time hand pose estimation using depth sensors, in: Consumer Depth Cameras for Computer Vision, Springer, 2013, pp. 119–137.

[26] D. Tang, H. Jin Chang, A. Tejani, T.-K. Kim, Latent regression forest: Structured estimation of 3d articulated hand posture, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3786–3793.

[27] J. Tompson, M. Stein, Y. Lecun, K. Perlin, Real-time continuous pose recovery of human hands using convolutional networks, ACM Transactions on Graphics (TOG) 33 (5) (2014) 169.

[28] L. Ge, H. Liang, J. Yuan, D. Thalmann, Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3593–3601.

[29] M. Oberweger, P. Wohlhart, V. Lepetit, Hands deep in deep learning for hand pose estimation, arXiv preprint arXiv:1502.06807.

[30] G. Poier, K. Roditakis, S. Schulter, D. Michel, H. Bischof, A. A. Argyros, Hybrid one-shot 3d hand pose estimation by exploiting uncertainties, arXiv preprint arXiv:1510.08039.

[31] C. Xu, L. Cheng, Efficient hand pose estimation from a single depth image, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 3456–3462.

[32] M. Dantone, J. Gall, G. Fanelli, L. Van Gool, Real-time facial feature detection using conditional regression forests, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 2578–2585.

[33] M. Sun, P. Kohli, J. Shotton, Conditional regression forests for human pose estimation, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 3394–3401.

[34] S. Hallman, C. C. Fowlkes, Oriented edge forests for boundary detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1732–1740.

[35] D. P. Solomatine, D. L. Shrestha, Adaboost. rt: a boosting algorithm for regression problems, in: Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on, Vol. 2, IEEE, 2004, pp. 1163–1168.

[36] J. H. Friedman, Greedy function approximation: a gradient boosting machine, Annals of statistics (2001) 1189–1232.

[37] S. Schulter, C. Leistner, P. Wohlhart, P. M. Roth, H. Bischof, Alternating regression forests for object detection and pose estimation, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 417–424.

[38] C. Li, K. M. Kitani, Pixel-level hand detection in ego-centric videos, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE, 2013, pp. 3570–3577.

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

[40] V. Vezhnevets, V. Sazonov, A. Andreeva, A survey on pixel-based skin color detection techniques, in: Proc. Graphicon, Vol. 3, Moscow, Russia, 2003, pp. 85–92.

[41] A. Criminisi, J. Shotton, Decision forests for computer vision and medical image analysis, Springer, 2013.

[42] S. Kullback, R. A. Leibler, On information and sufficiency, The Annals of Mathematical Statistics 22 (1) (1951) 79–86.