DISCRIMINATIVE POSE ESTIMATION USING MIXTURES OF GAUSSIAN PROCESSES

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Engineering and Physical Sciences

2013

By
Martin Fergie
School of Computer Science

Contents

Word Count: 999,999

List of Tables

List of Figures

Abstract

This thesis proposes novel algorithms for using Gaussian processes for Discriminative pose estimation. We overcome the traditional limitations of Gaussian processes, their $O(N^3)$ training complexity and their uni-modal predictive distribution by assembling them in a mixture of experts formulation [40]. Our first contribution shows that by creating a large number of fixed size Gaussian process experts, we can build a model that is able to scale to large data sets and accurately learn the multi-modal and non-linear mapping between image features and the subject's pose. We demonstrate that this model gives state of the art performance compared to other discriminative pose estimation techniques.

We then extend the model to automatically learn the size and location of each expert. Gaussian processes are able to accurately model non-linear functional regression problems where the output is given as a function of the input. However, when an individual Gaussian process is trained on data which contains multi-modalities, or varying levels of ambiguity, the Gaussian process is unable to accurately model the data. We propose a novel algorithm for learning the size and location of each expert in our mixture of Gaussian processes model to ensure that the training data of each expert matches the assumptions of a Gaussian process. We show that this model is able to out perform our previous mixture of Gaussian processes model.

Our final contribution is a dynamics framework for inferring a smooth sequence of pose estimates from a sequence of independent predictive distributions. Discriminative pose estimation infers the pose of each frame independently, leading to jittery tracking results. Our novel algorithm uses a model of human dynamics to infer a smooth path through a sequence of Gaussian mixture models as given by our mixture of Gaussian processes model. We show that our algorithm is able to smooth and correct some mistakes made by the appearance model alone, and outperform a baseline linear dynamical system.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Parts of this thesis have been published in the following:

- M. Fergie and A. Galata, Local Gaussian Processes for Pose Recognition from Noisy Inputs,
  British Machine Vision Conference 2010.
- M. Fergie and A. Galata, Dynamical Pose Filtering for Mixtures of Gaussian Processes,
  British Machine Vision Conference 2012.

Copyright

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the "Copyright") and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made **only** in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the thesis, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see `http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=487`), in any relevant Thesis restriction declarations deposited in the University Library, The University Library's regulations (see `http://www.manchester.ac.uk/library/aboutus/regulations`) and in The University's policy on presentation of Theses

Acknowledgements

I would like to thank my supervisor Aphrodite Galata for her guidance throughout my PhD.

My parents Josephine Coyne and Michael Fergie have given me tremendous support and encouragement throughout my education. Without their help this would not have been possible. I would also like to thank my partner Constance Laisné for her endless patience and love.

I would like to dedicate this work to my grand mother Brenda Fergie, who would have been very proud to have a 'doctor' in the family!

1 Introduction

Understanding human motion has seen a recent emergence into mainstream technology. Entertainment systems are increasingly bundled with interfaces which respond to human motion, allowing interfaces to move beyond buttons and menus. The most high profile example of such technology is Microsoft Kinect, a hardware and software platform that allows people to interact with computer gates using body motion. These interfaces are now being showcased in many areas of the entertainment industry. Other consumer devices such as televisions are increasing allowing human motion to control the television set through predefined gestures for changing channels or accessing and navigating menus.

As motion capture devices are making their way into peoples homes, there is an increasing need for applications which can make use of these systems. Applications such as video search, allowing users to search through videos of human motion using a motion itself as a search term. This opens up a plethora of opportunities for domains such as sign languages. These languages have evolved separately from spoken languages, developing an intricate grammar based on spatial locations and motion. The Internet forces deaf people to interact through a second language, a completely different form of expression. Effective motion estimation would permit deaf users to search television archives for phrases and subjects, it would permit sign language students to look up terms in a dictionary by performing the sign themselves.

Pose estimation technology would have a great effect on animation. Animated characters rely on human actors to make their expression convincing and life-like. However, capturing this human motion is expensive, requiring specialist motion capture systems of a time-consuming manual annotation process.

In this thesis we address pose estimation techniques that are fast, scalable and can be applied in varied environments. Particularly we focus on monocular pose estimation, where the pose is estimated from a single colour image. These techniques build flexible

models from off-line training corpora to allow pose estimation to be performed using individual images. This flexibility is ideal for tasks such as extracting pose from video archives such as YouTube or television sign language interpreters.

This is a very challenging problem. Models must be able to handle the high levels of ambiguity which is inherent when estimating 3D motion from 2D images. Sources of ambiguity include unobservable movement such as people moving towards and away from the camera, varying backgrounds, subject appearances and lighting conditions.

The techniques introduced in the thesis are designed to explicitly model these ambiguities, utilising probabilistic uncertainty to infer the correct pose in challenging conditions. They are also designed to be scalable, allowing large training corpora to be used, increasing the range and accuracy of these models.

1.1 MOTIVATIONS

In this thesis we contribute to the field of discriminative pose estimation, a family of techniques which learn a model for estimating human pose by modelling a mapping directly from the image evidence to the pose space. The need for these fast and flexible techniques is driven by a number of applications

- *Sign language analysis* — allowing native signers to interact with the Internet in a natural and intuitive manner. Creating the possibility for new applications such as searching by sign, dictionary look up or collaboratively edited documents.
- *Motion capture for animation* – giving amateur animators and film-makers the ability to capture human motions without requiring the expensive and time consuming methods currently available.
- *Video archives* – using motion capture devices to search on-line video archives. Flexible discriminative pose estimation methods can be used to crawl these archives extracting rich information about human motions and automatically categorising footage.

Pose estimation techniques for these applications need to be fast, to process large quantities of video, and be flexible enough to be deployed in a variety of environments.

1.2 CONTRIBUTIONS

The first contribution of this thesis is an appearance model for inferring human pose directly from images by using a mixture of Gaussian processes. We show how existing

mixture models can be adapted to facilitate the use of Gaussian processes (GP). The resulting model is able to overcome the limitation of GPs, their $O(N^3)$ training complexity and uni-modal predictive distribution. We demonstrate that this model outperforms state of the art techniques on human pose estimation data sets.

The second contribution builds on the first. We extend the model to jointly learn the size and location of each GP in the model. This allows each GP to model a region of the dataset that has coherent signal noise – a similar level of ambiguity. This has the effect of giving more accurate predictive distributions to each GP and ensuring that they model uni-modal regions of the data set. We show that this algorithm outperforms our original mixture of Gaussian processes model.

Our final contribution is a dynamics model which infers a smooth pose sequence for a video sequence. Discriminative pose estimation infers the pose for each frame individually. This results in a jittery tracking sequence, where the ambiguity in each estimate causes the predicted pose to jump around the true pose. We introduce a dynamics framework specifically for our mixture of Gaussian processes model which uses dynamic programming and a dynamical constraint to infer a smooth path through our predictive distribution.

1.3 THESIS OUTLINE

The rest of the thesis is structured as follows

Chapter 2 reviews the related work on human pose estimation. We cover a wide variety of approaches from the vast literature. We review which techniques are suitable for which circumstances and use that to motivate our choice of approach.

Chapter 3 sets out how we evaluate our contributions. We discuss how human pose is represented for pose estimation models. We cover the image features that we use to extract information from training images. Finally, we cover the different data sets that we used to evaluate our methods.

Chapter 4 gives the first of our contributions. We introduce a mixture of Gaussian processes model for discriminative human pose estimation. The proposed algorithm builds on the mixture of experts literature [40, 7] discussing how the linear experts of these models can make confident but incorrect predictions about the subject's pose. We then introduce Gaussian processes [59], showing how they are a powerful regression technique, but have their limitations when applied to large human pose estimation data sets. Finally, we show how our proposed algorithm is able to overcome these limitations to give state-of-the-art performance on discriminative human pose estimation data sets.

Chapter 5 extends the work from chapter 4 proposing an algorithm for optimising

the expert sizes and locations for a mixture of Gaussian processes model. We review the relevant techniques in the machine learning literature [61, 48] and discuss how their formulation restricts them from being applied to large data sets. We propose a novel algorithm incorporating some of these ideas which can be applied to human pose estimation data sets with a large number of training samples and high dimensional features.

Chapter 6 proposes an algorithm for combining a dynamical constraint with our mixture of Gaussian processes model. This algorithm uses dynamic programming techniques to infer a smooth path through the predictive distribution of each frame. While we introduce it in the context of our mixture of Gaussian processes model, this model can be used with any technique which has a mixture of Gaussian distributions as its predictive distribution.

Chapter 7 offers concluding remarks and suggests future directions for research.

2 Background

2.1 INTRODUCTION

Human pose estimation is tackled in a variety of approaches, each of which has distinct suitability for different applications. The early methods consisted of generative models which formulate the problem as a high dimensional search, generating pose hypotheses and evaluating their likelihood against the image evidence. These likelihood functions typically render a human body model into the image plane and then measure the dissimilarity between the projected human body model and the image evidence. This dissimilarity is interpreted as a probabilistic likelihood function and monte-carlo methods are used to optimise the likelihood function [38]. However, the high dimensionality of human pose makes naively sampling the pose space infeasible. As such, these methods rely on dynamics models and dimensionality reduction techniques in order to reduce the search space. These techniques are covered in section 2.1.1.

To overcome the reliance on densely sampling the pose space, discriminative models have been introduced which learn a direct mapping from the image evidence to the pose space [2]. These methods don't require an expensive sampling process, and are often much faster than the generative models outlined above. The mapping is learnt offline from a training set consisting of example images and their annotated pose. The mapping most commonly consists of a regression method that is able to model non-linear and multi-modal mapping from image to pose. This is the approach that we adopt in this thesis, as such we give a detailed overview of the literature in section 2.2.

Discriminative methods are limited by their reliance on a training set of pose annotated images, and they are unable to generalise to poses which don't lie in their training set. Pictorial structure models [27] attempt to overcome this limitation by modelling the appearance of individual body parts. They model human pose as a tree structured graph where each body part has a location and orientation, and are constrained by a

set of *springs*, modelling relative location of neighbouring body parts. Graph inference techniques are then used to find the correct pose. This approach allows their models to fit poses that lie outside the training set. However, they are limited to performing inference in the 2 dimensional image plane, and they require a large number of appearance model evaluations resulting in slow inference. We cover these techniques in section 2.1.2.

2.1.1 Generative Models

Generative models use a human body model rendered in the image plane to obtain an image likelihood. This measures how well a pose hypothesis fits the image evidence. For a pose hypothesis $\mathbf{y}$ and an image $\mathbf{x}$, the image likelihood represents the distribution $p(\mathbf{x}|\mathbf{y})$, the likelihood of the image evidence conditioned on the pose hypothesis. To find the optimal pose for an image, Bayes rule is used to obtain the distribution $p(\mathbf{y}|\mathbf{x})$:

$$p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{y})p(\mathbf{y})}{\int_{\mathbf{y}} p(\mathbf{x}|\mathbf{y})}. \tag{2.1}$$

Thus to find the optimal pose for an image, these models generate example poses $\mathbf{y}$ and compute their image likelihood $p(\mathbf{x}|\mathbf{y})$ to find the pose that maximises $p(\mathbf{y}|\mathbf{x})$. The image likelihood function $p(\mathbf{x}|\mathbf{y})$ is a multi-modal function as image ambiguities result in there being many poses that are well explained by the image evidence. This requires an exhaustive search of the pose space $\mathbf{y}$ as simple mode finding techniques will suffer from local minima. The high dimensionality of human pose makes this search computationally infeasible due to the large number of image likelihood evaluations. To make the search tractable, motion models and image cues are used to focus the search space to only plausible poses. These models can be broken into two categories, global models that sample poses from an accurate statistical distribution over the pose $p(\mathbf{y})$, and local models that perform a local search in the image space initialised with the previous pose estimate.

Global models require a compact representation of the pose distribution $p(\mathbf{y})$ which is able to model the multiple modes of the image likelihood function but remain compact enough to minimise the number of samples. This distribution is often given by a dynamics model which predicts the subject's pose given the pose estimates of the previous frames. These models give a distribution over the predicted pose $p(\mathbf{y}_t)$ at image frame t conditioned on the previous pose estimates $\mathbf{y}_{1:t-1}$. As such, the function that

generative models wish to maximise is

$$p(\mathbf{y}_t|\mathbf{x}_t) \propto p(\mathbf{x}_t|\mathbf{y}_t)p(\mathbf{y}_t|\mathbf{y}_{1:t-1}) \tag{2.2}$$

where the integral over $\mathbf{y}$ in the denominator of equation 2.1 is dropped. Estimating the pose for the first frame in the sequence $t = 1$ requires an expensive initialisation procedure.

A common technique for maximising this function is particle filtering [38] which represents the posterior distribution $p(\mathbf{y}_t|\mathbf{x}_t)$ using a set of weighted samples which are propagated from frame to frame. These weighted samples are able to represent and propagate the multiple modes of the likelihood function during tracking. The likelihood of each particle $\hat{\mathbf{y}}_\mathbf{t}$ is given by

$$p(\hat{\mathbf{y}}_t|\mathbf{x}_{1:t}) = p(\mathbf{x}_t|\hat{\mathbf{y}}_t) \int_{\hat{\mathbf{y}}_{t-1}} p(\hat{\mathbf{y}}_t|\hat{\mathbf{y}}_{t-1})p(\hat{\mathbf{y}}_t|\mathbf{x}_{1:t-1})d\hat{\mathbf{y}}_t. \tag{2.3}$$

The pose estimate for each frame can then be found by taking the expectation of the posterior distribution as modelled by the particles

$$\mathbf{y}_t = \mathbb{E}[p(\mathbf{y}_t|\mathbf{x}_t)] = \sum_{\hat{\mathbf{y}}_t} p(\hat{\mathbf{y}}_t|\mathbf{x}_{1:t})\hat{\mathbf{y}}_t. \tag{2.4}$$

Different particle sampling strategies have been explored such as annealing [24] or optimising individual particles [15, 14].

Local models formulate the tracking problem by taking a single pose estimate from the previous frame and propagating it using a dynamics model as above. However instead of maintaining a global set of samples or a distribution over the pose $p(\mathbf{y})$, the pose is optimised by taking a gradient of the image likelihood function $\frac{\delta p(\mathbf{x}|\mathbf{y})}{\delta \mathbf{y}}$ to obtain efficient search directions in the pose space [16, 79, 62].

An important component of both models is the image likelihood function which indicates how well a pose estimate matches the image evidence. The image likelihood function consists of two components, an articulated 3D human body model which can be projected into the image plane and a cost function for comparing the projected model to the image evidence. The human body model is typically represented as a hierarchy of body parts with fixed shape parameters. The pose of the model is represented using joint angles where each limb has 3 rotation parameters representing its rotation relative to the previous joint. The initial pose and shape parameters are either set manually [32] or obtained through an initialisation procedure [41]. The limbs are often represented by

primitive shapes such as cuboids [67], cylinders [72], ellipsoids [50, 35, 24] or tapered *superquadrics* [33, 79]. Other researchers have employed models that use realistic mesh-based body models [73, 32, 55]. Plankers et al. [56] use *metaballs* to model muscle and then fit a skin mesh over the top. Other techniques fit a mesh to a 3D visual hull and then learn a correspondence between each mesh point and each skeleton limb using *skinning* [32, 46, 89, 23]. Rosenhahn et al. [64] perform lower body tracking explicitly modelling skirts using a physics based cloth draping model. They show that they can accurately track the articulations of the legs despite being occluded by the skirt.

The image cost function is used to calculate how well the projected human body model configured with pose $\mathbf{y}$ matches the image evidence. These typically consist of matching the silhouette of the projected model to an image measurement. Image measurements used include, silhouette extraction [24, 73, 4, 21, 46], edge intensity images [24, 58] and volumetric reconstruction [35, 89, 32]. Global models tend to calculate a mismatch score between the projected model and the image evidence. Local models often require points in the image to be assigned to points on the projected model [79, 32]. This allows a gradient $\frac{\delta p(\mathbf{x}|\mathbf{y})}{\delta \mathbf{y}}$ to be derived using the Jacobian of the disparity between the model and the assigned image points. This gradient is then used to fit the model to the available image evidence.

As discussed above, approaches based on the particle filter algorithm [38] use dynamics models to propagate sample poses from previous frames to be evaluated against the new image frame. A poor dynamics model means that the model may *lose track* of the subject, requiring an expensive re-initialisation step. Common dynamic models include constant angular velocity [72] and Kalman filters [90, 80]. Brand [13] uses a hidden Markov model to learn a temporal model of human motion to perform 3D tracking where the human pose is modelled as a set of discrete states with Gaussian emissions. Brubaker et al. [18] build a dynamics model which simulates the physics of human walking. They model the physical forces acting on the legs, their centre of mass and a spring force to provide the walking motion. This model is used to drive their system for tracking side-on walking sequences.

By modelling human pose in a low dimensional manifold it is possible to reduce the search space and give accurate models of human dynamics which generalise well between different performances of a motion. The most common technique for modelling a human pose manifold is *principal component analysis* (PCA) [84, 8, 86] which finds the linear projection of the pose data such that the majority of the data's variance lies along a subset of the basis vectors called the *principal components*. The data is then represented using a subset of the available components that model the majority of the data's variance.

However, human pose contains lots of non-linearities which PCA is unable to model. Non-linear manifold techniques allow a lower dimensional manifold to be learnt while retaining high accuracy when mapped back into the full pose space. Non-linear techniques include *locally linear embedding* [66], ISOMAP [82] and GPLVM [35, 88, 26, 21]. The *Gaussian process dynamical model* (GPDM) learns a manifold where the latent points constrained temporally and spatially [87, 91]. This allows complex non-linear motion to be modelled using a low dimensional pose representation. To over come the $O(N^3)$ limitation of Gaussian processes, Chen et al. [21] manually segment their sequence and learn a separate GPDM for each subsequence, using a Markov model to select the GPDM during tracking. Hou et al. [35] learn a *back-constrained* GPLVM which ensures the latent points to have a similar spatial layout as the pose. These latent points are then clustered to form a discrete representation of the pose space. A *variable length Markov model* is then used to predict which clusters to generate particles from.

Another approach is to explore pose distribution in a top down fashion. Stenger et al. [81] model the pose distribution as a hierarchical tree, where the higher levels of the tree consist of broad image observations and the leaf nodes consist of detailed image observations. Inference proceeds by exploring the tree in a top down fashion, only exploring regions that have a high image likelihood from their parent nodes.

Local methods commonly rely on the tree structure of the human skeleton to simplify the pose search. Gavrila and Davis [33] fit their skeleton in a tree like fashion, first fitting the torso to the image evidence and then proceeding down each limb. Vlasic et al. [89] model a full surface mesh for their subject and break the tracking problem into to parts, skeleton estimation and surface estimation. They initialise their system with a mesh taken from a 3D laser scan [45] or a shape from silhouette technique [68] and a manually annotated skeleton. For each frame in the sequence, they fit their skeleton to the image evidence by deforming the surface mesh using linear blend skinning. They define an image cost function using silhouettes to build correspondences between the mesh points and image points allowing a least-squares cost formulation for optimising the skeleton. In the second stage, the mesh is deformed to match the silhouettes from each camera, this removes the artefacts caused by linear blend skinning. This method suffers from errors being propagated down the skeleton hierarchy, for example, if the upper arm is fitted incorrectly then the lower arm will also be fitted incorrectly. Gall et al. [32] overcome this by incorporating a global inference step when the system fails to find a limb. For example, if the lower arm does not match the image evidence sufficiently, then a global inference procedure will be used to fit the entire arm. This greatly reduces the requirement for manual intervention in their system. They also enrich the

image measurements by incorporating texture correspondences using the SIFT algorithm [47]. Liu et al. [46] have extended the above system to track two interacting characters as they perform activities such as dance and martial arts. They use a segmentation technique to assign each pixel to one of the characters, allowing each to be tracked individually.

The above techniques rely on using multiple cameras to resolve the depth ambiguity that results from a single view of human motion. These ambiguities cause singularities in the optimisation problem solved when fitting a 3D human body model to a single image. Morris and Rehg [51] introduce a scaled prismatic model which is a 2D representation of human pose. The 2D model does not suffer from the same singularities as a 3D model when fitting to the image data. Using offline training data, they learn a mapping from this 2D model to 3D human pose. Howe et al. [36] take a similar approach and use a mixture of factors model to map from their 2D tracking model to 3D pose.

Sminchisescu and Triggs [79] derive a second order image likelihood function allowing them to obtain a covariance representing the uncertainty over each pose variable. They use this covariance to focus their search along the dimensions which have greater uncertainty caused by image ambiguities. They also incorporate a number of model constraints directly into their optimisation problem. These constraints include hard joint angle limits, stabilisation priors, an inter-body penetration penalty and anthropometric shape priors.

2.1.2 Pictorial Structure Models

Pictorial structure models perform human pose estimation by optimising the location of individual body parts in the image plane. In a similar fashion as above, pictorial structure models maximise $p(\mathbf{y}|\mathbf{x})$ the probability of the pose $\mathbf{y}$ conditioned on the image evidence $\mathbf{x}$. However instead of generating samples from a prior over the pose as with a generative model, the inference problem is formulated as a tree-structured graph $G(V, E)$, where the vertices V are the body parts, and the edges E capture the spatial relationships between each body part. The objective function that they maximise is given by

$$p(\mathbf{y}|\mathbf{x}) \propto p(\mathbf{x}|\mathbf{y})p(\mathbf{y}) = \prod_i p(\mathbf{f}_i|\mathbf{y}_i) \prod_{(\mathbf{y}_i,\mathbf{y}_j)\in E} p(\mathbf{y}_i|\mathbf{y}_j) \tag{2.5}$$

where $\mathbf{y}_i$ refers to the parameters of an individual body part i, and $\mathbf{f}_i$ is an appearance model for that body part. The parameters for each body part $\mathbf{y}_i$ consist of the location and orientation of the body part. The appearance term, $p(\mathbf{f}_i|\mathbf{y}_i)$, captures how well the configuration of part i matches the image evidence. The pose prior, $p(\mathbf{y}_i|\mathbf{y}_j)$, models

the likelihood of body part i in position $\mathbf{y}_i$ given that body part j is in position $\mathbf{y}_j$. Dynamic programming techniques [8, 27] are used to maximise the objective function (2.5).

Ramanan [57] learn offline edge intensity templates for each body part and model the pose prior using distance histograms. This allows the pose model to represent a wide range of poses. Ferarri et al. [30] extend this model obtaining an initial location and scale for the head and torso using a human detector [22]. They then use this approximation to run grab-cut segmentation [65] to identify which image pixels belong to the subject and which belong to the background. A colour histogram appearance model is build from the foreground pixels, and they use a pictorial structures model as in [57]. They also experiment with adding spatial-temporal priors, using a previous image frame to initialise the pose search.

Andriluka et al. [3] model their pose prior using a Gaussian distribution by transforming the part locations into a space representing their relative angles. In this space the body part relationships are approximately Gaussian and it allows efficient inference using convolutions. Their appearance model consists of body part detectors using shape context descriptors extracted from sliding windows and an AdaBoost classifier [31].

Johnson and Everingham [39] introduce a mixture of pictorial structure models. They cluster the pose of their training set and represent each pose cluster with a different pictorial structure model. This allows their model to handle a wide range of poses. Their appearance model consists of a cascade of classifiers using HOG descriptors [22] and support vector machines [63].

Wang et al. [92] use the *Poselets* concept [12] to build an appearance model for their pictorial structure model. They learn a hierarchy of Poselets where each Poselet votes for a body part location. Their pictorial structure model then infers the location of each body part from the Poselet detections.

2.2 DISCRIMINATIVE HUMAN POSE ESTIMATION

Discriminative pose estimation attempts to directly model the mapping from image evidence to subject's pose $p(\mathbf{y}|\mathbf{x})$. This requires a training set of images with annotated pose to build an exemplar set. An image feature is extracted for each of the training examples which attempts to capture the information relevant to estimating the human pose. For an image feature $\mathbf{x}$ the predicted pose $\mathbf{y}$ is obtained using a mapping

$$p(\mathbf{y}|\mathbf{x}) = f(\mathbf{x}) \tag{2.6}$$

where $f()$ is a mapping learnt from the image features extracted from the training images $\mathbf{X} = \{\mathbf{x}_i\}_{n=1}^N$ and their annotated poses $\mathbf{Y} = \{\mathbf{y}_i\}_{n=1}^N$. However the mapping $f()$ is not strictly functional due to the ambiguities inherent in mapping directly from the image to the pose space. As such, $f()$ is often modelled by multi-modal regression techniques that are able model the ambiguous relationship between the image features and the subject's pose.

These methods work on each image frame individually, and so don't suffer from problems of losing track of the subject and requiring initialisation as with generative models. The mapping $f()$ also tends to be very fast at predicting the pose of a test image, in exchange for computationally demanding offline learning procedures. In this section we give an overview of research into the image features $\mathbf{x}$ used to capture the pose information, the models used to learn the mapping $p(\mathbf{y}|\mathbf{x})$ and finally models for incorporating dynamics into a discriminative framework.

2.2.1 Image Features

The image features used for discriminative pose estimation are largely derived from those used for object detection. One of the earliest attempts at discriminative pose estimation [2] extracted shape context descriptors [6] from silhouettes and clustered them using K-means. The cluster centres are used to form a codebook following a *bag-of-words* approach, where each entry captures a distinctive shape contained in the training images. To form an image feature the shape context descriptors extracted from each image are histogrammed with respect to the codebook. The histogramming is performed by centring a Gaussian on each codebook entry, allowing each extracted descriptor to vote softly with respect to the codebook entries. A relevance vector machine was then used to model the mapping between the histogrammed descriptors and the pose.

Ning et al. [53] introduce a novel descriptor inspired by the shape context descriptor that encodes local shape from image gradients as opposed to silhouette points. This allows their features to be applied to grey-scale images, removing the dependency on silhouette extraction. For each cell in the descriptor they extract the dominant edge orientation and its magnitude to build a representation of local shape. These descriptors are then clustered along with their (x, y) locations in the training images to form a codebook. Instead of centring a Gaussian on each codebook, they learn a Mahalabonis distance which represents the relevance of the individual descriptor components. A Bayesian mixture of experts model [40] is used to learn the mapping to the pose space.

Sminchisescu et al. [78] evaluate two alternative features built from SIFT descriptors [47]. The first feature is constructed by densely sampling SIFT descriptors from a

bounding box around the subject to form a 8064D vector. The second feature clusters extracted SIFT descriptors to form a bag of words model as outlined above. They train their method on pseudo-synthetic images, where a synthetic human is rendered on to real images. They show that the densely sampled feature outperforms the bag-of-words model when there are cluttered backgrounds using their Bayesian mixture of experts model. They suggest that this is because the background clutter is distributed among the useful information with the bag-of-words feature. With the densely sampled feature, the useful information and clutter are represented in different feature vectors, and the learning model is able to separate them through learning linear weights for each feature. The disadvantage of this feature is that the high dimensionality will cause difficulties for some regression techniques.

Kanaujia et al. [42] review a collection of hierarchical features, where low level image cues are repeatedly combined to produce robust image features. These features include HMAX [69] and *hyperfeatures* [1]. HMAX features are biologically inspired features which are designed to match the visual cortex of primates. They use a set of manually selected Gabor filters to extract low-level image features. These features are then pooled using hierarchical max operations to obtain a low dimensional representation. Hyperfeatures use a hierarchy of local bag of words models to produce scale and translation invariant features. At the base level, SIFT descriptors are densely extracted for the entire image. These descriptors are clustered in local regions to form a codebook for each portion of the image at level n of the hierarchy. The features for each spatial block within a region are histogrammed with respect to the codebook to create a set of higher level features. This process repeats, using the histogrammed features from level n to produce a codebook for level $n + 1$. The features extracted from multiple levels can then be combined to form a single feature vector.

Kanaujia et al. [42] show that these features are effective for human pose estimation tasks when using a mixture of experts model. In order to extract relevant information from their features they experiment with canonical correlation analysis (CCA) [70] and relevant component analysis (RCA) [5]. CCA learns a set of linear basis functions that project their features and pose data to maximise their joint correlation. This has the effect selecting relevant features for their pose prediction. RCA is used to learn a Mahalabonis distance to give the distance between two feature vectors $d(\mathbf{x}, \mathbf{x}')$ as an unnormalised Gaussian with covariance D. The diagonal values of D represent the relevance of the individual features of $\mathbf{x}$ to the pairwise distance. They show that utilising these techniques are able to reduce the sensitivity to background clutter.

Extracting silhouettes has shown to be a very useful technique for improving the

accuracy of discriminative pose estimation [2, 49, 11]. However, accurate silhouette extraction is often difficult and requires a background model. Poor quality silhouettes can strongly degrade the pose estimation accuracy. Ionescu et al. [37] incorporate silhouette extraction into the pose estimation procedure. They use the *constrained parametric min-cuts* segmentation algorithm [20] to obtain ranked segmentations of the human from the background. The ranking is given by a learnt quality function which scores how accurate each candidate segmentation is. The segmentation quality function $g()$ is jointly learnt alongside a regression function $f()$ to map the feature extracted from the segmentation to the subject's pose. The predicted pose is then obtained by choosing the segment that maximises the segment quality function, and using that silhouette to predict the pose

$$\mathbf{y} = f(\underset{i}{\operatorname{argmax}}\, g(\mathbf{x}_i)) \tag{2.7}$$

where $\mathbf{x}_i$ denotes a feature extracted from silhouette i. They show that this method is able to accurately estimate the pose of humans in cluttered scenes.

The recent availability of structured light depth cameras through the Microsoft Kinect platform has enabled the use of depth features [71, 34]. These cameras give a very accurate depth image allowing easy background segmentation, operation in dark environments and invariance to subject colour variations. Depth features use pixel wise depth comparisons to represent the local 3D structure of the image. The features extracted for each pixel, p, are the set of depth disparities of neighbouring pixels selected using random offsets from p. These features are used to train a random forest classifier which classifies each pixel into one of 32 body parts. The output of the classifier is a probability map for each body part giving the likelihood distribution over its location on the surface of the subject. Mean-shift clustering is used to locate each body part's position from its probability maps.

2.2.2 *Appearance Models*

The mapping from the extracted image features to 3D pose is multi-modal due to ambiguities in inferring 3D pose from 2D images. It also contains large amount of noise caused by shadows, clothing and noisy backgrounds. As such, appearance models must be able to represent these properties in order to give an accurate predictive distribution over the pose.

Agarwal and Triggs [2] evaluate the use of regression methods based on sparse linear models. These models give a prediction for the pose $\mathbf{y}$ as a linear function of the image

features $\mathbf{x}$

$$\mathbf{y} = \mathbf{W}^T \phi(\mathbf{x}) + \mathbf{b} + \epsilon \tag{2.8}$$

where $\mathbf{w}$ is a set of learnt weights, $\phi(\mathbf{x})$ is a feature transformation function, $\mathbf{b}$ is a constant bias and ϵ is a noise process. In its simplest form, the feature transformation function such that $\phi(\mathbf{x}) = \mathbf{x}$ which gives $\mathbf{y}$ as a linear function of $\mathbf{x}$. To model a non-linear mapping a kernel function $k(\mathbf{x}_1, \mathbf{x}_2)$ can be used, in which case the predictive function for the i^{th} pose variable is

$$y_i = \sum_{n=1}^{N} w_{n,i} k(\mathbf{x}, \mathbf{x}_n) + b_i + \epsilon \tag{2.9}$$

where $\mathbf{x}$ is the test feature and $\mathbf{x}_n$ are the training examples. A kernel function can take on many forms, but it typically represents the distance between $\mathbf{x}_1$ and $\mathbf{x}_2$ as an inner product. In this formulation the weights $w_{n,i}$ control the influence of each training example on the prediction.

Sparse linear models such as the *support vector machine* (SVM) and *relevance vector machine* (RVM) place priors on the weights to push them towards zero. In the linear formulation, this has the effect of removing irrelevant features from the pose prediction. In the kernel formulation, this has the effect of selecting only the most relevant training examples for pose prediction, allowing for very fast inference. Agarwal and Triggs [2] evaluate both linear and kernel models where the weights are learnt using standard least squares, a SVM and a RVM and show that kernel SVM gives the best performance for 3D human pose estimation. They show that a RVM is able to give very similar performance but with much sparser solutions, only requiring 6% of training examples, compared to 53% with a SVM.

These models have the limitation that they only give one predictive mode, their predictive distribution is typically Gaussian, with the mean and variance given by $\mathcal{N}(\mathbf{y}|\mathbf{w}^T\phi(\mathbf{x}), \Sigma)$ where Σ is given by the noise process ϵ in equation 2.8. A *mixture of experts* model [40, 7] overcomes this issue by modelling the predictive distribution as a weighted combination of K linear models

$$p(\mathbf{y}) = \sum_{i=1}^{K} \pi_i \mathcal{N}(\mathbf{y}|\mathbf{w}_i^T \phi(\mathbf{x}), \Sigma_i) \tag{2.10}$$

where π_i is a weight assigned to each expert using a logistic regression model. These models have received a lot of attention in human pose estimation [76, 78, 43, 42, 11, 53, 83]. The experts can consist of linear models [53, 78], relevance vector machines

[42, 83] or support vector machines [42]. Kanaujia and Metaxas evaluate different formulations for learning the expert weights π_i including *iterative re-weighted least squares* and *Bayesian multi-category classification*. Sminchisescu et al. [78] use a generative human body model to evaluate the weights for each expert. They show that when combined with densely sampled SIFT descriptors, incorporating the generative model leads to a significant improvement in pose estimation accuracy.

The main limitation with the kernel based linear models is that selecting the form of the kernel function must be done through cross-validation. In order to train these models, kernel parameters along with regularisation parameters must be chosen to get a good trade off between data fit and generalisation. Another limitation is the formulation of how the predictive variance, Σ, is obtained. When a test point is sufficiently far from selected training examples used as basis functions, the predictive variance collapses towards zero which can lead to confident but incorrect predictions.

A Gaussian process (GP) [59] overcomes these limitations by giving a likelihood over the kernel function. This allows kernel parameters to be optimised using gradient based optimisation techniques. This also enables the use of kernels where each dimension of the input features can be given a different kernel length scale, allowing relevant features to be selected. Further, the predictive variance does not collapse towards zero when the test feature lies outside of the training set. However, a GP can only represent a unimodal function and requires $O(N^3)$ training time to optimise the kernel parameters. This limits their use to small data sets.

Zhao et al. [93] show that a GP can give comparable performance to a mixture of experts model on human pose estimation. To overcome the limitations outlined above, Urtasun and Darrell [85] construct online local Gaussian processes centred around each test point. When predicting the pose of an unseen test point, they construct a GP for the K nearest training points $\vartheta_i, i = 1, \ldots, K$. Each online model is constructed from the S points closest to ϑ_i in the pose space. The parameters for the model are taken from a set of Gaussian processes trained offline on local regions of the data set. The prediction is given as a mixture of Gaussian distributions, where the prior for each component is computed as a function of its inverse variance. By constructing online local Gaussian processes, the model is able to scale to large data sets and model multi-modal mappings. However, using the predictive variance to set the component weights causes the model to bias its predictions to the modes with less uncertainty in the data. The online model construction also makes test inference slow as it requires computing and inverting a new kernel matrix for each online test point.

Bo and Sminchisescu [10] propose *twin Gaussian processes* as a way of learning the

structure of the outputs. Gaussian processes only give a prediction over a single output variable, so for pose estimation an individual GP is learnt for each joint [85, 93]. This has the disadvantage that the prediction of each joint is independent, possibly resulting in a prediction where one joint doesn't match the rest of the subject's pose. Bo and Sminchisescu jointly learn two Gaussian processes, one to model the pose predictions and one to model the structure between each of the joint predictions. This has the effect of enforcing the model to predict entire poses, rather than independent joints. They show that this model outperforms a single GP on the HumanEva data set [74].

Ek et al. [26] use a *Gaussian process latent variable model* (GPLVM) to perform human pose estimation as a relational mapping between the image features and the pose. A GPLVM [44] learns a set of latent points, $\mathbf{Z}$, which form a non-linear low-dimensional representation of a variable, $\mathbf{Y}$, by optimising a GP likelihood $p(\mathbf{Z}, \mathbf{Y}|\theta_{Z\to Y})$. The latent points $\mathbf{Z}$ are jointly optimised along with the GP parameters $\theta_{Z\to Y}$ which represent the mapping from $\mathbf{Z} \to \mathbf{Y}$. Ek et al. [26] learn a shared latent space, where each latent point $\mathbf{z}_n \in \mathbf{Z}$ is associated with a training image feature $\mathbf{x}_n$ and its corresponding pose $\mathbf{y}_n$. The latent space is constrained such that each latent point corresponds to a single point in the pose space, forcing all the multi-modality to be captured in the mapping from the latent space to the image feature space. Inferring the pose from a test image consists of finding the latent point that maximises the GP likelihood of the latent point to the observed feature

$$\hat{\mathbf{z}} = \underset{\mathbf{z}_*}{\operatorname{argmax}}\, p(\mathbf{x}_*|\mathbf{z}_*, \mathbf{X}, \mathbf{Z}, \theta_{Z\to X}) \tag{2.11}$$

where $\mathbf{x}_*$ is the test feature, $\mathbf{z}_*$ is a latent point and $\theta_{Z\to X}$ are the GP parameters for the mapping from latent to feature space. The corresponding pose, $\hat{\mathbf{y}}_*$, is then found by mapping the optimal latent point $\hat{\mathbf{z}}$ into the pose space using the Gaussian process prediction

$$\hat{\mathbf{y}}_* = \mathbb{E}_{\mathbf{y}_*}\left[p(\mathbf{y}_*|\hat{\mathbf{z}}, \mathbf{Z}, \mathbf{Y}, \theta_{Z\to Y})\right]. \tag{2.12}$$

While this technique is able to model the multi-modal mapping from image to pose, it still requires learning Gaussian process models to represent the entire data set, resulting $O(N^3)$ training complexity, and limiting it to small data sets.

Memisevic et al. [49] take a similar approach using *shared kernel information embedding* (SKIE) to learn a shared latent space between the image features and the pose. They represent the features $\mathbf{X}$, pose $\mathbf{Y}$ and latent points $\mathbf{Z}$ using Gaussian kernel densities.

For example the density over the feature space is given by

$$k_X(\mathbf{x}, \mathbf{x}') = \exp\Big\{ \frac{(\mathbf{x} - \mathbf{x}')^2}{2\sigma_X^2} \Big\}. \tag{2.13}$$

To map from the feature space $\mathbf{X}$ to the latent space $\mathbf{Y}$ a conditional distribution is derived from the corresponding kernel densities

$$p(\mathbf{z}|\mathbf{x}) = \sum_{n=1}^{N} \frac{k_X(\mathbf{x}, \mathbf{x}_n)}{\sum_{j=1}^{N} k_X(\mathbf{x}, \mathbf{x}_j)} k_Z(\mathbf{z}, \mathbf{z}_n). \tag{2.14}$$

which gives the probability of observing a latent point $\mathbf{z}$ given a feature $\mathbf{x}$. This is a bi-directional mapping, allowing conditional densities to be derived between ($\mathbf{X}$, $\mathbf{Z}$) and ($\mathbf{Y}$, $\mathbf{Z}$) in both directions. The kernel parameters representing each space σ_X and σ_Y are selected using the average nearest neighbour distance in the training data. The kernel parameter for the latent space σ_Z is arbitrary as it just has the effect of scaling the entire latent space.

Learning in the model requires optimising the set of latent points $\mathbf{z}_n \in \mathbf{Z}$ to maximise the regularised joint mutual information between the two mappings

$$\hat{\mathbf{Z}} = \underset{\mathbf{Z}}{\operatorname{argmax}} \Big\{ I(\mathbf{X}, \mathbf{Z}) + I(\mathbf{Y}, \mathbf{Z}) + \frac{\lambda}{N} \sum_{n=1}^{N} ||\mathbf{z}_n||^2 \Big\} \tag{2.15}$$

where λ controls the amount of regularisation. This is set using an annealing procedure, initialised as a large value and decreased as the model is trained. The value that minimises the error on a validation set is chosen for test inference. Test inference uses the same principal as the shared GPLVM of Ek et al. [26]. First the most likely latent point $\hat{\mathbf{z}}$ is identified by maximising $p(\mathbf{z}|\mathbf{x}_*)$. This latent point is then used to find its corresponding pose $\hat{\mathbf{y}} = \operatorname{argmax}_{\mathbf{y}_*} p(\mathbf{y}_*|\mathbf{x}_*)$.

Their approach is $O(N^2)$ compared to $O(N^3)$ for the GPLVM, allowing it to be applied to much larger data sets. They demonstrate the use of online local models centred around each test point. These models take the 25 nearest neighbours in the input space and learn a local SKIE online. They show that both local and global models out perform GPLVM on the Poser data set [2].

Shotton et al. [71] take a segmentation based approach to discriminative pose estimation. By using depth images obtained from the Microsoft Kinect structured light camera, they approach the pose estimation problem by attempting to segment individual body parts in the 3D surface image. They identify 32 body parts which represent

small regions of the body and use a random forest classifier to segment each of these body parts from a depth image.

Using depth features, they obtain a probability of each pixel belonging to each body part by using a random forest classifier [17]. A random forest classifier is an ensemble of binary decision tree classifiers. Each tree classifier consists of split nodes and leaf nodes. Split nodes use a test function $f(\mathbf{x}, \lambda) \rightarrow \{0, 1\}$ to propagate an input feature, $\mathbf{x}$, down to the next layer of a binary tree by selecting its left or right sub-tree based on the parameter λ. These functions can take on a number of forms, the most common is an axis-aligned split which applies a threshold (λ) to a subset of the features in $\mathbf{x}$. The leaf nodes contain stored predictions over the target variable. In the classification case, they typically store class histograms which give the probability for a pixel reaching that leaf node belonging to each class.

Decision tree classifiers are well known to over-fit their training data, creating models that don't generalise well to test data. Random forests train an ensemble of trees where each tree is trained on a random permutation of the original training data and a limited parameter set. The prediction from a random forest is made by taking the average over each individual tree's prediction.

Shotton et al. [71] train their decision trees by learning split functions which threshold the depth disparities at each pixel. The trees are trained in a hierarchical manner, recursively selecting the best parameter that splits the data at each node in the tree up to a fixed depth. Each pixel of each training image is assigned a class label denoting which body part it belongs to and the set of all pixels are used to train each tree.

To obtain a pose estimate, a skeleton is fitted to the identified body parts by offsetting the joints by a fixed distance into the depth image. This method performs pose inference in the image space in a similar fashion to a pictorial structures model. This means that unlike other discriminative models described in this section, it is unable to infer a 3D pose from a 2D image.

Girshick et al. [34] extend the above to use regression forests. Instead of storing a set of class probabilities at each leaf node, they store a vote towards a 3D joint position. The votes are given as 3D offsets from the classified pixel. They use the same classification metric as Shotton et al. [71] to learn the structure of the trees. The hough voting procedure similar to [52] is used to obtain a set of candidate joint locations from the votes. Using regression forests in this way achieves greater accuracy with less training samples and gives faster test inference. This technique has also been applied to 2D upper body pose estimation on RGB images using colour histogram features instead of depth disparities [54].

These methods [71, 34, 52, 54] infer the joint positions in the image space, requiring 3D depth images to infer 3D pose. While it would be possible for [34, 52] to infer 3D pose from a 2D image, none of the above papers evaluate how well this performs. As such, they can't be directly compared to the other discriminative techniques outlined above [2, 49, 11, 85], which infer the 3D pose from a 2D image.

2.2.3 Dynamics Models

Dynamics models are not as common in discriminative pose estimation as they are not so strictly required as with generative models, and they are more difficult to combine with discriminative appearance models.

Agarwal and Triggs [2] learn a second order autoregressive dynamics process to resolve the ambiguities in their appearance model. They combine the dynamics and appearance by conditioning their appearance model on both the image features and the propagated pose. This has the effect of resolving the multi-modality of the appearance model by conditioning it on a predicted pose propagated from a previous frame. They integrate this model into the CONDENSATION [38] framework by using their discriminative predictive distribution to evaluate particle likelihoods. It should be noted that this differs to the traditional generative usage of the CONDENSATION framework where particles are evaluated using an expensive image likelihood function. Instead particles are evaluated using the Gaussian prediction of the combined appearance and dynamics prediction from the discriminative regression model. They show that this model is able to improve their tracking performance compared to a discriminative appearance model alone.

Thayananthan et al. [83] use a model that works in a similar fashion to a particle filter, where they maintain L pose estimates which are propagated using Kalman filters. At each frame, they obtain K pose estimates from a bank of relevance vector machines conditioned on the image features. These are combined with the dynamical predictions to form $L \times K$ predictions for each frame. A generative style image likelihood function is then used to weight each prediction and the best L predictions are then propagated to the next frame. This model uses a discriminative model to predict a set of pose hypotheses for each frame, and then a generative image likelihood model to select which hypothesis are propagated. The image likelihood model needs to be specifically developed for each data set.

Sminchisescu et al. [77] introduce a probabilistic framework for incorporating a dynamics prediction with a discriminative appearance model. They use a mixture of experts model which is conditioned on both the image features and the previous pose.

This gives a predictive distribution over the pose as a mixture of Gaussian distributions conditioned on the image feature and the previous pose estimate

$$p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{t-1}) = \sum_{i=1}^{K} \pi_i \mathcal{N}(\mathbf{y}|\mu_i(\mathbf{x}_t, \mathbf{y}_{t-1}), \Sigma_i(\mathbf{x}_t, \mathbf{y}_{t-1})), \tag{2.16}$$

where $\mu_i(\mathbf{x}_t, \mathbf{y}_{t-1})$ and $\Sigma_i(\mathbf{x}_t, \mathbf{y}_{t-1})$ are given by a Bayesian mixture of experts model [40, 77]. To incorporate this into a dynamics framework, they maintain L pose hypotheses for each frame. To predict a new frame, they make a prediction using their mixture of experts model conditioned on the image feature $\mathbf{x}_{t+1}$ and the L pose hypotheses for $\mathbf{y}_t$. This results in $L \times K$ Gaussian predictions which are clustered using a variational approximation scheme [75] to obtain L predictions for frame $t + 1$. They demonstrate that this dynamics framework improves their tracking results on some of their more complex sequences.

2.3 DISCUSSION

In this chapter we have given an overview of techniques for human pose estimation. Generative models which explicitly model the appearance of the subject perform well in controlled environments and tend to be the best at generalising to new poses. They infer the pose by rendering pose proposals into the image space and evaluating an image likelihood function. This function is often expensive to evaluate and is typically the bottleneck of such systems due to the high number of required evaluations. Dynamics models are of crucial importance to give a compact proposal distribution. When deployed in a monocular setting, the problem is less well formulated leading to a more complex pose distribution. These models have an even higher dependence on dynamical models and other constraints in order to keep the problem tractable.

Pictorial structure models attempt to estimate the 2D pose of the subject in the (x, y) image space. They model human pose as a tree-structured graph where each body part has a discrete location and rotation. Graph inference techniques are used to optimise an objective function which captures the likelihood of each part at a particular image location and orientation and the relationships between neighbouring body parts. These models are able to perform strongly in unconstrained environments with noisy and cluttered backgrounds. However they are only able to estimate poses encountered in an offline training set and are computationally expensive to fit.

Discriminative models, as used in this thesis, attempt to model the subject's pose directly by learning a mapping from an image feature to the pose space. These methods

use an offline training set to learn a model which directly maps from an extracted image feature to the pose space. These models are typically very fast for inferring the pose of a test image, and are able to operate on individual images, without requiring dynamics models. While these techniques don't require a dynamics model, it has been shown that incorporating a dynamics constraint can improve tracking in some situations. The main limitation of discriminative models is their dependence on a training set consisting of pose annotated images. They are unable to infer the pose of images which contain unseen poses. Microsoft Kinect has shown that with a large corpora of training data and reliable image features discriminative techniques can be used in a commercial motion capture setting.

3 Data Representation

In this chapter we discuss the data representation and problem setting which we use to evaluate the methods contributed in this thesis. We look at the problem of discriminative pose estimation – estimating 3D human pose directly from a monocular image. In §3.1 we show how human pose can be represented in a form suitable for monocular human pose estimation. §3.2 covers the image features and segmentation techniques that are used to evaluate our models and finally §3.3 covers the data sets used for evaluation.

3.1 REPRESENTING HUMAN POSE

We wish to model articulated human pose, that is the position of all the limbs of a person as they perform an activity. The pose can be broken down into two components, the global position and orientation of the skeleton, and the articulated configuration of the skeleton. For representing the articulation of the skeleton there are two dominant representations, joint angles – where a kinematic skeleton is constructed and the pose is represented by a 3-dimensional rotation at each joint, and joint positions – where each joint represents a 3-dimensional vector offset from an origin in 3D space.

The joint angles approach has the advantage that a pose expressed as joint angles is independent of intra-subject variations such as limb-lengths and height. A given pose can be mapped onto skeletons with different proportions, still expressing the same pose. This is a valuable property for modelling human actions, where the action modelled should be relatively independent of the person performing it. In the context of discriminative pose estimation, a kinematic skeleton can have undesirable properties due to errors propagating down the kinematic tree. For example, if there is an error in the estimated joint rotation for the shoulder joint, this error effects the position of all joints on that arm. This effect has a large impact on applications which require the 3D position of the hand, such as manipulating virtual objects.

We take the same approach to representing human pose as Memisevic et al. [49] and represent the pose as joint positions, a 3-dimensional vector encoding the joint's location relative to subject's *root joint*. This representation is very flexible as it does not require a skeleton to be built for each subject. Instead each joint is represented independently, removing the problem of errors propagating down the kinematic tree. This is a natural choice for discriminative pose estimation for which a large proportion of models estimate the location of each joint independently [53, 11, 77, 42, 85].

While suitable for pose estimation, this representation is less suitable for building generic gesture models for recognising and categorising human gestures. Fortunately converting between joint positions and joint angles can be performed in a closed form, allowing the pose estimated in joint positions to be converted to joint angles, and vice-versa.

We select the base of the subject's spine as the root joint, and rotate the joint positions into a coordinate frame that is independent of the camera viewpoint. This is performed by using the camera's extrinsic parameters to rotate the joints into a coordinate frame where the X and Y axes lie in the image plane, and the Z axis represents the depth of the joint from the camera. This allows images from multiple cameras to be combined to learn models from a large training set of different subjects from different camera angles.

This representation has been chosen for it's relative simplicity and ease of interpretation. The models used in this paper are independent of the pose representation, and discriminative pose estimation techniques have shown to be effective on a wide range of human pose representations [2, 77, 53, 49]. We model the 3D articulated pose of the subject centred around their root joint. We assume that the global location of the human in the image is obtained using a human detector. This allows our trained pose estimation model to be invariant of the location of a human in the scene.

For the purposes of our learning algorithms, the i^{th} joint can be represented as a 3D vector $\mathbf{y}_i = [x_i, y_i, z_i]$. We concatenate these vectors to form a complete pose vector $\mathbf{y} = [\mathbf{y}_1, \ldots, \mathbf{y}_J]$ to represent the entire human pose.

3.1.1 *Smoothing Hand Annotated Pose*

Many pose estimation data sets have ground truth information which has been annotated by hand. This gives a noisy pose signal caused by the inconsistencies in the annotator's placement of the joints in each image. We smooth the pose signal of each joint using a linear low-pass filter [1]. This removes the high-frequency jitter from the pose

[1] `intfilt` from the MATLAB package, using parameters $l = 4$, $p = 4$ and $\alpha = 0.3$.

signal, leading to visually smoother pose annotations. The cut-off frequency is manually selected to smooth the pose data without losing the articulation of the underlying pose.

3.1.2 *Calculating Pose Estimation Errors*

To evaluate pose estimation models we need to calculate an error measurement that reflects how close a predicted pose $\mathbf{x}'$ is to the ground truth pose $\mathbf{x}$. In this thesis we follow [11] and use the mean absolute error (MAE) given by

$$\text{MAE} = \sum_{i}^{D} |x_i - x'_i|. \tag{3.1}$$

where i indexes the individual components of each joint's position, D is the total number of dimensions in the pose representation. This gives the estimation error for a single frame in the sequence. To compare the performance of different models we compute this error metric for each frame, and then give the mean and standard error over the entire sequence. The units of the metric are either given in millimetres or pixels, depending on the underlying pose representation of the data set. The standard error is calculated as

$$\text{SE} = \frac{\sigma}{\sqrt{n}} \tag{3.2}$$

where σ is the standard deviation of the mean absolute error of a sequence, and n is the number of frames in the sequence.

3.2 IMAGE FEATURES

For discriminative human pose estimation, an image representation is required that is able to capture information relevant to discriminating the pose of the subject, while suppressing background noise and maintaining a relatively low dimensionality. This is performed by extracting an image feature which applies a transformation to the image signal, followed by some high level processing. The features used are commonly adapted from the object detection literature, which shares common requirements. In this section we discuss the image features used in this thesis and discuss their relative merits.

3.2.1 *Bag of Words*

There exist a number of local image descriptors that represent the local shape surrounding around a point of interest in an image. These descriptors include SIFT [47], *shape*

context [6] and *histogram of oriented gradient* (HOG) [22] descriptors. Each of these descriptors represent an image patch using a fixed size vector. SIFT and HOG descriptors represent a histogram of edge orientations in spatial cells located around a centre. These can be applied directly to a grey scale or colour image to obtain a representation of the edge information in that image patch. Shape context descriptors represent histograms of contour points in log-polar cells distributed around a point of interest, and are typically applied to silhouette images [2].

These descriptors represent local image patches which represent the local shape around a central point. To build a representation for an entire image of a human we must combine descriptors sampled at many locations in the image. A simple approach for performing this is to densely sample the descriptors on a regular grid [22, 78, 10]. However choosing the size of each descriptor leads to a trade-off between having features of very high dimensionality (8064D in [78]) or having a course image representation.

To overcome this problem, a *bag-of-words* model builds a histogram of local descriptor responses to represent each image. The histogram bins consist of a *codebook* – a set of exemplar descriptors that are obtained by clustering a large set of descriptors extracted from a set of training images. For a set of N training images $I_{1:N}$, the descriptor set $\mathbf{d}^{(n)}$ extracted from each image are concatenated and clustered using the K-means algorithm. Each of the K cluster centres are used to form the codebook entries, $\mathbf{C} = \{\mathbf{c}_i\}_{i=1}^{K}$. Each entry $\mathbf{c}_i$ represents a distinctive and informative shape observed in the training set.

An image feature $\mathbf{x}_n$ can be computed by histogramming the descriptors $\mathbf{d}^{(n)}$ extracted from image n with respect to the codebook $\mathbf{C}$. To construct an image feature, we first assign each descriptor in $\mathbf{d}^{(n)}$ to it's nearest codebook entry. We represent this using an index vector $\mathbf{d}'$ whose j^{th} element is given by

$$\mathbf{d}'_j = \underset{i \in D}{\operatorname{argmin}} |\mathbf{d}_j^{(n)} - \mathbf{c}_i|, \tag{3.3}$$

where $\mathbf{d}_j^{(n)}$ is the j^{th} descriptor from image n. The i^{th} element of the image feature $\mathbf{x}_n$ is then computed as the number of descriptors which are closest to codebook entry $\mathbf{c}_i$

$$x_{n,i} = \left|\{i : j \in \mathbf{d}', i = j\}\right| \tag{3.4}$$

Each image feature $\mathbf{x}_n$ is normalised, $\sum_{i \in K} x_{n,i} = 1$, such that the feature is invariant to the number of descriptors extracted from each image. Image features are computed for test images using an analogous process, using the same codebook $\mathbf{C}$ extracted from

the training set.

The focus of this thesis is on the learning models to infer the pose, as such we construct our bag of words features following the configuration of other researchers [49, 11, 2]. The local descriptors that we extract for each image depend on the data set. For the Ballet and HumanEva data sets where silhouettes can be extracted we use shape context descriptors sampled from every point on the silhouette contour [2, 11]. For the sign language data set the dynamic background prevents background subtraction from being performed. As such, we use SIFT descriptors sampled from interest points identified using the difference-of-Gaussian technique from the original SIFT algorithm [47]. Using this technique the number of descriptors is determined by the default SIFT algorithm parameters.

To construct the code book, we use K-means clustering [2, 53, 11] and create a code-book consisting of 300 entries following [49, 11]. For large data sets, the number of descriptors extracted from all training images can grow too large for clustering. In such situations we follow [49] and cluster 40,000 descriptors randomly sampled from the complete set.

3.2.2 Hierarchical Features

Another popular class of features hierarchically pool information from low level features to form high level descriptive features that offer a degree of invariance to low level noise [42]. In this thesis we use the HMAX feature which is a biologically inspired model built to represent the visual cortex of primates. It has shown to give very high performance for both object recognition [69] and human pose estimation [85, 42].

The HMAX features are built from a hierarchy of alternating sets of simple and complex cells. The simple cells, (S1, S2) are filter operations which extract information from the image layers, and the complex cells, (C1, C2) pool this information using a local max operation to give some local scale and transformation invariance. The layers used for feature computation are applied in the order S1, C1, S2, C2. The process is illustrated in figure 3.1.

The S1 layer consists of a bank of Gabor filters at a range of scales and orientations which have been selected using the same stimuli used to probe biological neurons [69]. These filters are arranged into 8 scale bands, with each band containing filters of two sub-scales and 4 orientations. For example, the first scale band contains filter sizes of 7 and 9 pixels at 4 orientations. The second scale band contains filter sizes of 11 and 13 pixels. The input image is processed with all filters from all scale bands to give a set of response images.

The C1 layer is formed by taking local maxima from the responses of the S1 layer. The response for a pixel in scale band i and orientation θ in the C1 layer is computed as the maximum value of S1 taken from the sub-scale responses in band i, over a local grid of size Σ_i. This has the effect of giving the output of the C1 units invariance to local transformation and scale effects. The output of the C1 layer consists of the local maxima response images for the 8 filter bands at 4 orientations.

The S2 layers randomly select a set of fixed size patches from the C1 responses at all scales and orientations to form the centres of RBF basis functions. These filters are applied to each scale band and orientation to obtain a set of response images for each patch. We use patches of sizes 8×8, 12×12 and 16×16 and extract 100 (K in figure 3.1) patches per size resulting in 300 patches in total.

Finally, the C2 layer performs a maximum over the orientations, positions and scale for each patch from the S2 layer. The output from each scale band is concatenated to form a 300D feature vector describing the image. In a similar fashion to the bag of words features above, each feature element corresponds to a local shape observed in the training images. The magnitude of its value is proportional to the strength of the shape's observation in the current image frame.

3.2.3 Background Subtraction

Many discriminative pose estimation techniques extract the subject's silhouette to isolate background image information from the subject [2, 11, 49]. This is performed using motion segmentation techniques, where the subject's movement is used to identify which pixels belong to the subject, and which pixels belong to the background. In this thesis, we evaluate our models both with and without background segmentation.

We use the background segmentation technique of Zivkovic [94] as implemented in OpenCV[2] which uses Gaussian mixture models (GMM) to build a model of the background colour distributions. The probability distribution of a pixels colour is given by by a K component GMM

$$p(p_i) = \sum_{k=1}^{K} \pi_k \mathcal{N}(p_i|\mu_k, \sigma_k \mathbf{I}) \tag{3.5}$$

where μ_k and σ_k are the mean and variance of each component. The mixture components are estimated online as new images are observed. The model maintains a set of the T most recent images $I_{t-T:t}$ which are used to train the GMM parameters at each pixel.

[2] http://opencv.willowgarage.com/wiki/

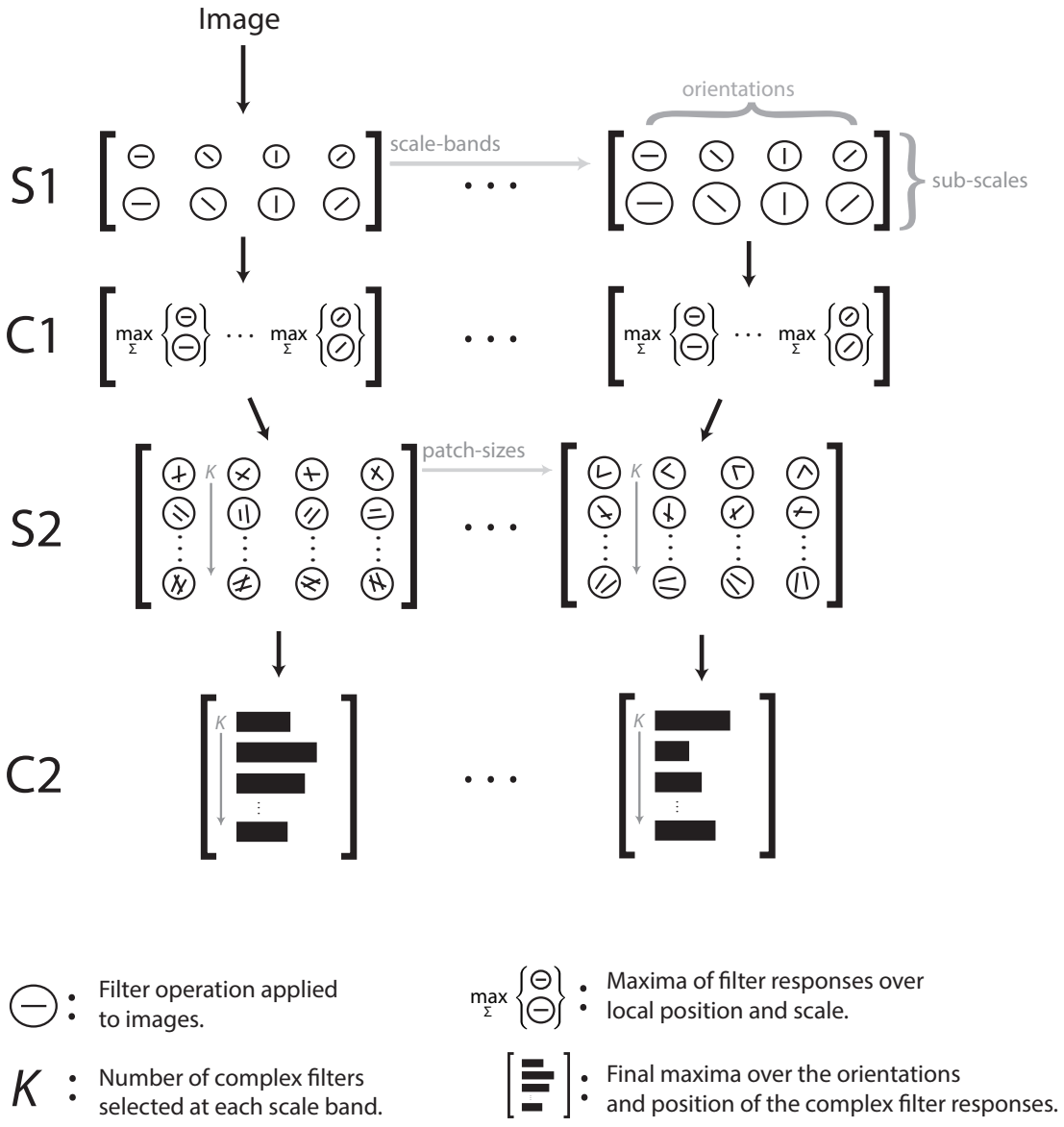

Figure 3.1: Illustration of HMAX features. Image is processed through a hierarchy of simple filter layers, S1, S2 and complex maximisation layers, C1, C2. This yields features that have high invariance to local scale and rotation. See text for details.

Typically the model is initialised using a sequence of empty frames such that $p(p_i)$ models the background colour distribution. When a foreground object enters p_i indicated by a significant change in the pixels colour, the model creates a new GMM component with parameters π_k, μ_{k+1} and σ_{k+1}. The aim of the algorithm is to detect this new colour as belonging to a separate foreground object. This is done by modelling the background using the L components with largest mixing priors

$$p(p_i|B) = \sum_{l=1}^{L} \pi_{\vartheta_l} \mathcal{N}(p_i|\mu_{\vartheta_l}, \sigma_{\vartheta_l}\mathbf{I}) \tag{3.6}$$

where ϑ stores a list of component indices sorted in descending order of π_k.

This model makes an online judgement about whether each pixel belongs to the foreground or background. It works on the premise that each pixel belongs to the background for the majority of the sequence, and is able to detect a foreground object when the pixels colour changes. This gives it the property that if a foreground object enters the scene and remains static for a sufficient period of time, it will eventually become the dominant component of (3.6) and be modelled as background.

To segment our sequences we initialise the above model on a sequence of frames of a empty scene without the subject present. This allows the mixture models at each pixel to learn the background colour distribution. The subject can then be detected as foreground when they enter the scene. To achieve satisfactory results we use a Gaussian filter to smooth our images before running the background subtraction and apply morphology techniques to close internal holes in the silhouette. To ensure that our silhouette represents the subject alone, we follow [74] and use connected components analysis to select the largest foreground object as our silhouette. Silhouettes are manually checked for each sequence to ensure that this process has the correct effect.

3.3 DATA SETS

We use three data sets to evaluate our contributions in this thesis. The first is a Ballet data set [35] which has been recorded and annotated in the School of Computer Science, University of Manchester, a sign language data set captured by [19] and HumanEva, a publicly available data set for human pose estimation [74].

3.3.1 Ballet

The Ballet data set consists of 5 repetitions of a complex ballet choreography performed by the same dancer. The data set is captured on 5 cameras which have been used to

Figure 3.2: Images from the Ballet data set. Left shows an example image, and the right shows it's extracted silhouette.

obtain annotated 3D joint positions for the subject. We use 4 of the sequences for training, and the final sequence for testing. This results in 1601 training frames and 356 test frames.

To prepare the joint positions for our pose estimation model, we compute a root joint at the base of the spine by taking the average position of the hip joints for each leg. The other joints are then given as offsets from the root joint. The extrinsic camera parameters supplied with the data set are then used to rotate the coordinates into the camera coordinate frame as described in §3.1. As this data set has been hand annotated, we smooth the pose signal as described in §3.1.1.

We use the background subtraction technique in §3.2.3 to obtain silhouettes for the data set. The model is initialised on approximately 20 video frames where the scene is empty. Figure 3.2 shows an image from the data set along with it's extracted silhouette. As the subject's movement can stretch to fill the majority of the frame, we don't apply any crop to the images before extracting the image feature.

3.3.2 Sign Language

The sign language data set consists of 6000 frames of footage taken from BBC television of a signer interpreting the news. This is a challenging data set due to the fast movement challenging dynamic background. As this data set has been captured from a television broadcast, there is no correct partitioning of the data where we have a training and test set which contain the same behaviours. We evaluate our models by breaking the sequence into chunks of 400 frames, and then randomly selecting partitions of these chunks to form our train and test partitions. We evaluate the models on 5 random partitions to reduce any bias introduced by the specific partitioning. The resulting data sets have 4400 training frames, and 1200 test frames.

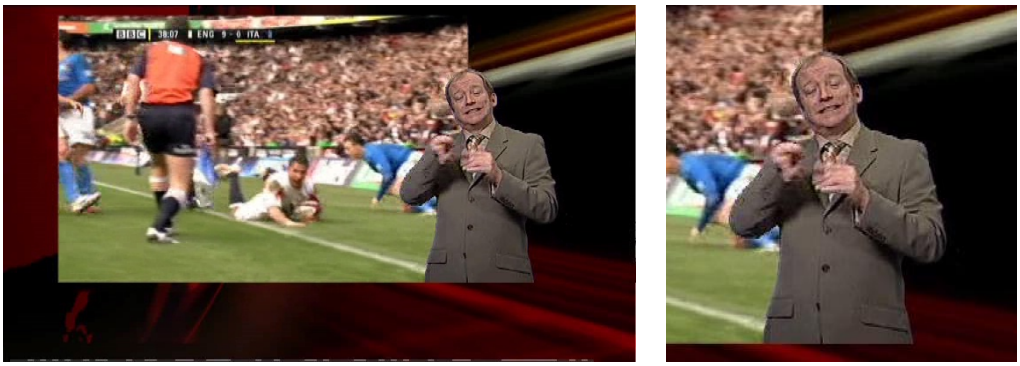

Figure 3.3: Image from the sign language data set. Left shows an example image, and the right shows the cropped image from which we extract the image feature.

The ground truth annotations are represented as 2D joint positions expressed as image pixel locations. We model the position of the head, shoulders, elbows, wrists and the tip of the hands. Due to the fixed location of the signer, we are able to crop the image such that they are centred and we represent the pose as pixel locations in this window. As this data set has been hand annotated, we smooth the pose signal as described in §3.1.1. An image from the data set can be seen in figure 3.3.

Due to the dynamic background, we are unable to use the background segmentation technique described in §3.2.3. Instead the image features are extracted directly from the grey scale image. For the bag-of-words feature, we sample SIFT descriptors following the different of Gaussian interest point selection in original SIFT algorithm [47].

3.3.3 HumanEva

The HumanEva data set [74] contains multiple subjects and multiple activities and has been captured specifically for the task of evaluating pose estimation systems. Sequences are captured using 3 colour cameras and 4 grey scale cameras. Pose is given as 3D joint positions obtained using a commercial motion capture system. The data set comes with predefined train, validation and test partitions. Since the test data is not made publicly available, we follow other researchers [11] and use the validation set as test data. To train models, we gather the data from subjects 1, 2 and 3 using the images from camera 3. Camera 3 is chosen as it is the camera from which we can recover the most reliable silhouettes.

The pose is represented as joint positions relative to the base of the subject's spine. These joint positions are then rotated to be in the camera's coordinate frame as in the Ballet data set. We extract the image feature from a bounding box around the subject to minimise the effect of background features. This bounding box can be obtained using

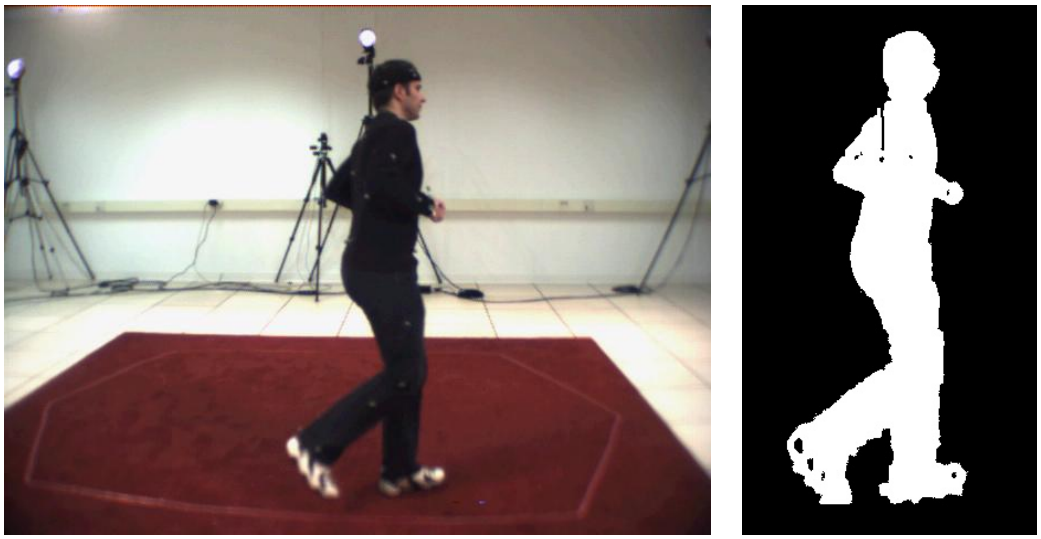

Figure 3.4: Image from the Jog sequence of the HumanEva data set. Left shows an example image, and the right shows a cropped silhouette image from which we extract our features.

the boundaries of the silhouette. Figure 3.4 shows an example image from the data set and the extracted silhouette. The background subtraction can be unreliable on the HumanEva data set due to the noisy images and subtle variations in the background and lighting. Figure 3.5 gives some example silhouettes where the background subtraction has failed. In some cases, this leads to image features extracted from silhouette data to give poor results compared to those extracted directly from the image.

3.4 DISCUSSION

In this chapter we have discussed the specific details on how we represent human pose while providing some discussion on alternative methods. We have covered the image features that we used to extract the image information relevant to human pose. Finally, we have discussed the three pose estimation data sets that are used to evaluate our models.

Figure 3.5: Example silhouettes from the HumanEva data set where the background subtraction shows significant errors. This can lead to poor pose estimation performance.

Mixture of Gaussian Processes 4

In this chapter we show how Gaussian processes can be used for performing human pose estimation in a mixture of experts framework. We start by giving a review of mixture of experts models and their use in human pose estimation. In §4.2 we introduce Gaussian process regression demonstrating how it solves some of the issues with the regression models used in mixture of experts models. In §4.3 we introduce our model for using Gaussian processes in a mixture of experts setting. This demonstrates how we can overcome the limitations of Gaussian processes such that they can be applied in a human pose estimation setting. Parts of this work have been published in [29].

4.1 BAYESIAN MIXTURE OF EXPERTS

Discriminative human pose estimation presents a difficult problem for regression techniques. Noise and ambiguity in the image features present the learning model with a multi-modal and non-linear regression problem. This means that standard linear regression techniques where the pose $\mathbf{y}$ is given as a linear function of the features $\mathbf{x}$ and a set of learnt weights $\mathbf{w}$ are unable to learn an accurate mapping. A Bayesian mixture of experts model uses a mixture of linear models to model multi-modal problems. Each expert is an individual linear model that is learnt on a local region of the data set. The predictive distribution of a Bayesian mixture of experts model is given by

$$p(\mathbf{y}|\mathbf{x},\theta) = \sum_{i=1}^{K} p(z=i|\mathbf{x})\mathcal{N}(\mu_i(\mathbf{x}),\Sigma_i(\mathbf{x})). \tag{4.1}$$

where K is the number of experts, $\mu_i()$ and $\Sigma_i()$ are the mean and variance of the prediction and are given by a Bayesian linear regression model

$$\mu_i(\mathbf{x}) = \mathbf{w}_i^T \phi(\mathbf{x}), \tag{4.2}$$

$$\Sigma_i(\mathbf{x}) = \beta^{-1} + \phi(\mathbf{x})^T S_i \phi(\mathbf{x}), \tag{4.3}$$

where S_i is the precision parameter of a Gaussian prior placed on the weights $\mathbf{w}$, β is a precision parameter of the outputs $\mathbf{y}$ and $\phi()$ represents a *basis function*. Each input vector $\mathbf{x}$ will typically have a '1' concatenated onto the end such that the corresponding weight acts as a bias, allowing the model to represent outputs that aren't centred around zero. The role of the basis function is to project the features $\mathbf{x}$ into an alternative feature space to extend the model to being able to model non-linear functions. An example basis function is a n-degree polynomial function, $\phi(\mathbf{x}) = [\mathbf{x}, \mathbf{x}^2, \ldots, \mathbf{x}^n]$ where the resulting vector is formed by concatenating increasing powers of $\mathbf{x}$. This allows the linear model to fit polynomial functions. For pose estimation it is common to use *radial basis functions* where each function is centred on a training point and $\phi_n(\mathbf{x})$ gives the distance of $\mathbf{x}$ to the n^{th} training point. Basis functions of this form leads to kernel models which we describe in §4.1.2.

Each expert is given a weight $p(z = i|\mathbf{x})$ using a logistic regression model. This model selects which experts to use when predicting test points. This is a linear classification model which maps an input feature $\mathbf{x}$ into a set of class probabilities $p(z|\mathbf{x})$. The probability that a feature $\mathbf{x}$ belongs to class i is given by

$$p(z = i|\mathbf{x}) = \frac{e^{\mathbf{v}_i^T \mathbf{x}}}{\sum_{j=1}^{K} e^{\mathbf{v}_j^T \mathbf{x}}} \tag{4.4}$$

where $\mathbf{v}_i$ is a set of weights learnt for class i.

4.1.1 Learning a Bayesian Mixture of Experts

Learning a Bayesian mixture of experts model involves learning each of the individual expert models and the logistic regression model to weight the expert contributions. Figure 4.1 shows a graphical model for a mixture of experts model. The algorithm learns two sets of weights, the expert weights, $\mathbf{W}$, and the weights for the logistic regression model $\mathbf{V}$. We use $\theta = \{\mathbf{W}, \mathbf{V}, \boldsymbol{\alpha}, \beta\}$ to collectively denote all model parameters. A Gaussian prior with zero mean and variance α_i is placed on the weights of each expert,

$$p(\mathbf{w}_i|\alpha_i) = \mathcal{N}(\mathbf{w}_i|0, \alpha_i^{-1}).$$

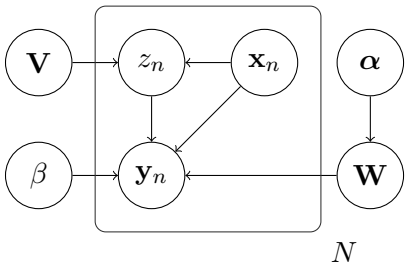

Figure 4.1: Graphical model for a Bayesian mixture of experts.

This prior acts as a regularisation term on the expert weights and is necessary to avoid over fitting [8, 7].

To optimise the model parameters, a latent variable $\mathbf{z} = \{z_n\}_{n=1}^{N}$ is used to assign each training point n to belong to an expert i. This variable can be encoded as a $N \times K$ matrix, where $z_{n,i}$ gives the probability that training point n belongs to expert i. To represent probabilities all values of $\mathbf{z}$ must lie between 0 and 1 and the rows should sum to 1

$$\begin{aligned} &0 \leq z_{n,i} \leq 1, \\ &\sum_{i \in K} z_{n,i} = 1 \; \forall n \in N. \end{aligned} \tag{4.5}$$

The *expectation maximisation* (EM) algorithm is used to optimise the model parameters $\mathbf{W}$ and $\mathbf{V}$. This algorithm iterates between two steps. The *E-step* optimises the latent variables $\mathbf{z}$ with respect to the current value of the model parameters θ, and the *M-step* then uses the updated $\mathbf{z}$ to maximise the likelihood of the model parameters θ. This is repeated for a fixed number of iterations or until the model's marginal likelihood converges as illustrated in algorithm 1.

The latent variables $\mathbf{z}$ are computed by taking the expectation of their distribution with respect to the rest of the model parameters, thus each value $\mathbf{z}_{n,i}$ is calculated as the likelihood of point n belonging to expert i

$$z_{n,i} = p(z_n = i, \mathbf{x}_n, \theta) p(y_n, z_n = i | x_n, \theta), \tag{4.6}$$

$$= p(z_n = i, \mathbf{x}_n, \theta) \mathcal{N}(y_n | \mu_i(\mathbf{x}), \Sigma_i(\mathbf{x})). \tag{4.7}$$

The first term on the right hand side is the probability assigned using the logistic regression model in (4.4). These likelihoods are normalised for each training point as described in (4.5).

The model parameters θ are then optimised with respect to the marginal likelihood

of the model. This is performed using standard techniques for optimising linear models, except that the influence of each training point is varied for each expert using $\mathbf{z}$. For example, the optimal weights for each expert are found using

$$\mathbf{w}_i = S_i^{-1} \sum_{n=1} z_{n,i} \mathbf{x}_n y_n, \tag{4.8}$$

where S_i is the precision of the prior on the weights, for full derivation see [25, 7]. Thus the latent variables $\mathbf{z}_i = \{z_{n,j} : n \in N, j = i\}$ allow each expert to only model a local region of the data set. This is an important property of the EM algorithm that restricts it being directly applied to Gaussian process experts, this will be covered in §4.3.

Algorithm 1 Expectation maximisation algorithm. M represents the desired number of iterations.

for all iter $= 1 \rightarrow M$ **do**
  $\mathbf{z} \leftarrow \mathbb{E}[p(\mathbf{z}|\mathbf{X}, \mathbf{Y}, \theta)]$
  $\theta' \leftarrow \text{argmax}_\theta \, p(\mathbf{Y}|\mathbf{X}, \mathbf{z}, \theta)$
  $\theta \leftarrow \theta'$
end for

4.1.2 Kernel Expert Models

Basis functions allow each expert to model a non-linear mapping by making a non-linear projection of the input features $\mathbf{x}$. It is common in human pose estimation tasks to use radial basis functions which centre a Gaussian distribution on each training point. Features are then expressed as a set of distances for an input $\mathbf{x}_*$ to each of the training points $\mathbf{x}_n$, $\forall n \in N$. The predictive mean of the linear model can be expressed as a sum over n basis functions

$$\mu_i(\mathbf{x}_*) = \sum_{n=1}^{N} w_n^T \phi_n(\mathbf{x}_*), \tag{4.9}$$

where we have defined $\phi_n()$ to be a Gaussian basis function centred around training point $\mathbf{x}_n$. This can also be expressed using a kernel function

$$k(\mathbf{x}, \mathbf{x}') = e^{-\frac{(\mathbf{x}-\mathbf{x}')^2}{2\sigma^2}} \tag{4.10}$$

which gives the distance between two points $\mathbf{x}$ and $\mathbf{x}'$ as a Gaussian density. The kernel parameter σ controls the variance of the kernel and is learnt using *cross validation*.

Cross validation is a technique for estimating the optimal model parameters using a *validation set* taken from the training data. For a given model parameter, the model is

trained using the training data with the validation set removed. The validation set is the used to estimate the accuracy of the model with the selected parameter. This process is repeated for a set of candidate model parameters, and the best performing parameter from the selection is chosen. It is common to use K-fold cross validation, where the training data is divided into K partitions. The model parameter is then evaluated with each K partitions used as the validation set. This ensures that all data points are used in both the training and validation sets.

The model parameters evaluated are typically selected from a regular grid in parameter space. In order to achieve a reasonable exploration of the parameter space, a large number of parameters must be evaluated. This leads to an expensive training procedure, as the model will typically have to be trained hundreds of times, using different parameters and validation sets. To make matters worse, some kernel linear models, such as the support vector machine, require a regularisation parameter to be selected using cross validation. These parameters are not independent, and require all their mutual combinations to be evaluated.

Once the model parameters have been selected, the weights of the best performing model are used to make a prediction

$$\mu(\mathbf{x}_*) = \sum_{n=1}^{N} w_n k(\mathbf{x}_*, \mathbf{x}_n) + b \tag{4.11}$$

where we have added an explicit bias parameter b. Using a kernel model changes the test prediction complexity to be $O(ND)$, where N is the number of training samples and D is the dimensionality of the feature space $\mathbf{x}$. For large data sets, this can have a dramatic impact on the computational cost of prediction compared to using a normal linear model where $\phi(\mathbf{x}) = \mathbf{x}$ which is $O(D)$. Bayesian mixture of expert models using input kernels have been used by many researchers for human pose estimation [77, 43]. However, the $O(N^2)$ prediction time limits their scalability to large data sets. As a result, other researchers have used sparse linear models as experts such as the *relevance vector machine* (RVM).

A RVM is able to build a sparse model by learning which training examples are the most relevant for test inference. In the linear model with kernel basis functions described above, the influence of each training point $\mathbf{x}_n$ is determined by the value of its corresponding weight $\mathbf{w}_n$. A RVM gives an individual Gaussian prior $\mathcal{N}(w_n|0, \alpha_n)$ to each weight $\mathbf{w}_n$. This has the effect of pushing many of the weight parameters w_n to zero, essentially pruning training data from the model, allowing for sparse predictions to be made. This greatly reduces the computational demands of predicting test data, as

only a small number of training samples are retained for making predictions – these are known as the *relevant vectors*. Thayananthan et al. [83] adapt the RVM for use in a multivariate mixture of experts setting and apply their model to human pose estimation.

However these models have the undesirable property that as the test point $\mathbf{x}_*$ moves away from the training points $\mathbf{x}_n$ the certainty of the prediction increases. As stated in (4.3) the variance of a linear model is given by

$$\Sigma_i(\mathbf{x}_*) = \beta^{-1} + \phi(\mathbf{x}_*)^T S_i \phi(\mathbf{x}_*). \tag{4.12}$$

This is the sum of a fixed precision parameter β and a variance term that depends on the basis function $\phi_n(\mathbf{x}_*)$ responses and the weights precision S_i. However, as a test input lies far from the training data, the output of the basis functions $\phi_n(\mathbf{x}_*)$ decreases towards to zero causing the predictive variance to decrease. This means that the model can give confident but incorrect predictions. In this work we investigate into the use of Gaussian processes which is a non-linear regression technique that does not suffer from these problems.

4.2 GAUSSIAN PROCESS REGRESSION

A Gaussian process is an extension of a linear model, where instead of defining a Gaussian prior over the weights, a Gaussian prior is placed over the functions $\mu(\mathbf{x}, \mathbf{w})$. Thus, the training data $\mathbf{x}$ and weights $\mathbf{w}$ are no longer model parameters, instead a Gaussian process defines a distribution over functions to which they are parameters. In this section we show how Gaussian process regression can be derived from the perspective of the linear model outlined above. For the purpose of introducing Gaussian processes we denote y as a single output variable, in §4.3 we explain how we use Gaussian processes with multiple outputs. If we consider a linear model as a function $f(\mathbf{x})$, the prediction over a pose variable y is given by

$$y = f(\mathbf{x}) + \epsilon, \tag{4.13}$$

where $f(\mathbf{x})$ is given as

$$f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}), \tag{4.14}$$

and ϵ is a random noise process. This is analogous to the linear model prediction in (4.2). For a set of training data, consisting of inputs $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$ and outputs $\mathbf{y} = \{y_n\}_{n=1}^N$,

the prediction can be rewritten as follows

$$\mathbf{y} = \Phi\mathbf{w}.$$

where Φ denotes a *design matrix* $\Phi_{n,i} = \phi_i(\mathbf{x}_n)$. We can obtain a Gaussian prior over the function values $\mathbf{f} = \{f(\mathbf{x}_n)\}_{n=1}^N$ observing

$$\mathbb{E}[\mathbf{f}] = \Phi\mathbb{E}[\mathbf{w}] = 0 \tag{4.15}$$

$$\text{cov}[\mathbf{f}] = \mathbb{E}[\mathbf{f}\mathbf{f}^T] = \Phi\mathbb{E}[\mathbf{w}\mathbf{w}^T]\Phi^T = \frac{1}{\alpha}\Phi\Phi^T = \mathbf{K} \tag{4.16}$$

where K is a gram matrix whose elements consist of kernel function evaluations $\mathbf{K}_{n,m} = k(\mathbf{x}_n, \mathbf{x}_m)$. This result is obtained by placing the same Gaussian prior on the weights as in a linear model, $\mathcal{N}(\mathbf{w}|0, \alpha^{-1})$. As such, a Gaussian process prior is placed on $\mathbf{f}$ giving a Gaussian distribution with zero mean and covariance given by the kernel function

$$p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{f}|0, K) \tag{4.17}$$

This distribution can be extended to include an unseen test point $\mathbf{x}_*$

$$\begin{bmatrix}\mathbf{f}\\ \mathbf{f}_*\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\mathbf{f}\\ \mathbf{f}_*\end{bmatrix}\middle|\mathbf{0}, \begin{bmatrix}\mathbf{K} & k(\mathbf{x}_*, \mathbf{X})\\ k(\mathbf{x}_*, \mathbf{X})^T & k(\mathbf{x}_*, \mathbf{x}_*)\end{bmatrix}\right) \tag{4.18}$$

This models the joint distribution of the training points and the unseen test point. From this we can obtain a Gaussian predictive distribution over the pose variable y_*

$$p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}, \theta) = \mathcal{N}(y_*|\mu(\mathbf{x}_*), \sigma(\mathbf{x}_*)) \tag{4.19}$$

where the mean and variance are given by

$$\mu(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{X})\mathbf{K}^{-1}\mathbf{y}, \tag{4.20}$$

$$\sigma(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - k(\mathbf{x}_*, \mathbf{X})^T\mathbf{K}^{-1}k(\mathbf{x}_*, \mathbf{X}). \tag{4.21}$$

A Gaussian process gives a Gaussian prediction over the outputs y as a function of the test input $\mathbf{x}$, training data $(\mathbf{X}, \mathbf{y})$, and a kernel function $k(\mathbf{x}, \mathbf{x}')$. This is fundamentally different to the linear models outlined above, where the model learns a set of weights as a parameter to a fixed set of basis functions $\phi_i(\mathbf{x})$. Gaussian process learning takes the form of optimising the kernel *hyper-parameters* which will be covered in §4.2.2.

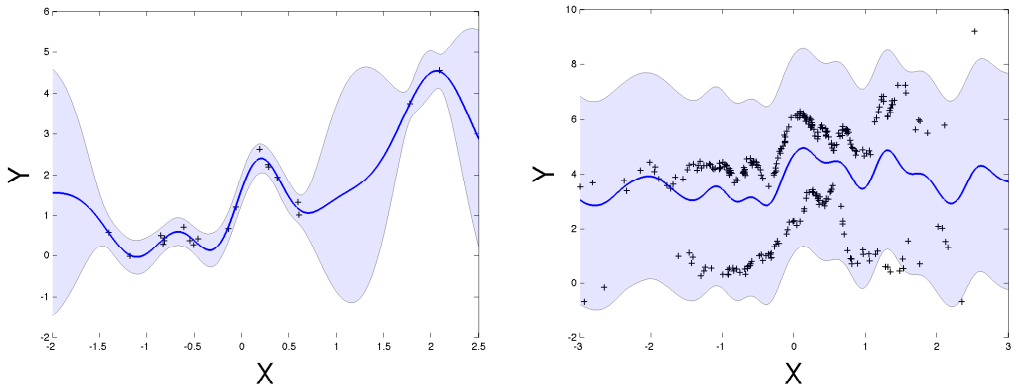

Figure 4.2: Gaussian process regression. Black crosses indicate training data, the blue line represents the mean and variance of the predictive distribution. (a) demonstrates a GP modelling a uni-modal function, (b) shows a multi-modal function where the Gaussian process averages the two modes.

The predictive mean in (4.20) can be thought of as a weighted average of the training outputs where the weights are given by the kernel distance between the training data and the test point. The predictive variance can be interpreted such that the first term $k(\mathbf{x}_*, \mathbf{x}_*)$ captures the variance inherent in the data and the second term incorporates the information from the neighbouring training data. When there is closely neighbouring training data, the second term is larger resulting in a more certain prediction. If there is no neighbouring training data, this term will fall to zero and the predictive variance is given by $k(\mathbf{x}_*, \mathbf{x}_*)$. This offers a major advantage compared to normal linear models including the RVM [9]. As discussed above, these have a variance formulation such that as the test input $\mathbf{x}_*$ moves away from the training data, the predictive variance decreases. Figure 4.2(a) shows an example of Gaussian process regression on synthetic data. The predictive distribution accurately models the data and gives higher variance in regions where there is no training data to constrain the predictive distribution.

4.2.1 The Kernel Function

The kernel function represents the similarity between training examples which allows the model to make predictions about test data. The kernel function can take on many forms, Rasmussen and Williams [59] give a detailed overview of the different kernel

functions available. In our work we use kernels based on a squared exponential function

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2_{signal} \exp\left(-\frac{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{P}^{-1} (\mathbf{x}_i - \mathbf{x}_j)}{2} \right) + \mathbf{b} + \sigma^2_{noise}\delta_{ij}, \qquad (4.22)$$

where the kernel *hyper-parameters* are $\theta = \{\mathbf{P}, \sigma^2_{signal}, \mathbf{b}, \sigma^2_{noise}\}$, σ^2_{signal} is the signal noise, $\mathbf{b}$ is a constant bias and σ^2_{noise} is a noise term. P represents the bandwidth of the kernel, this parameter adjusts how quickly the output variable y varies with respect to the input $\mathbf{x}$. We evaluate two formulations, an isotropic kernel $\mathbf{P} = \mathbf{I}p$ where p is the bandwidth applied to all input dimensions, and a kernel with automatic relevance detection $\mathbf{P} = \mathrm{diag}([p_1, \ldots, p_D])$ where a bandwidth is learnt for each input dimension. By learning a bandwidth for each input dimension, the model is able to detect which dimensions of the input features are most relevant for pose prediction – enabling the model to prune out noisy image features.

4.2.2 *Gaussian Process Learning*

The kernel hyper-parameters θ can be optimised by maximising the marginal log likelihood with respect to the training data $\mathbf{y}, \mathbf{X}$

$$\log p(\mathbf{y}|\mathbf{X}, \theta) = -\frac{1}{2}\mathbf{y}^T\mathbf{K}^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}| - \frac{n}{2}\log 2\pi. \qquad (4.23)$$

This computation is dominated by computing the inverse of the kernel matrix, $\mathbf{K}^{-1}$, which is $O(N^3)$. Partial derivatives $\frac{\delta}{\delta\theta_j} p(\mathbf{y}|\mathbf{X}, \theta)$ can be computed allowing each of the kernel parameters to be optimised using gradient descent. This allows kernel parameters to be accurately learnt from the entire training set, without having to rely on cross validation as is commonly used with other kernel regression models including the RVM.

4.2.3 *Limitations*

The main limitations of Gaussian processes is their cubic computational complexity for training. The predictive distribution given in (4.20) and (4.21) depends on $\mathbf{K}^{-1}$, the inverted kernel matrix. Inverting a $N \times N$ matrix is $O(N^3)$. The computational cost of inferring a test point is $O(N^2)$. To keep these computational costs manageable, we learn multiple small Gaussian processes such that each model contains approximately 100 points.

As with normal linear models they can only model uni-modal functions. Figure 4.2(b) demonstrates a Gaussian process trained on a multi-modal function, where there

are two possible output modes for each input. This causes the model to average over the two modes resulting in a poor fit. This type of multi-modality appears in human pose estimation where ambiguous poses, such as side-on walking, result in multiple output poses for an image feature. In the next section we show how Gaussian processes can be incorporated into a mixture of experts framework to handle such occurrences, resulting in a novel model that can accurately estimate the predictive distribution of human pose given an image feature.

4.3 MIXTURE OF LOCAL GAUSSIAN PROCESSES

In this section we demonstrate how Gaussian processes can be used for human pose estimation, constructing a model consisting of multiple Gaussian process *experts* each modelling a local region of the data set. This allows the model to handle multi-modal mappings, as well as overcoming the computational limitations of Gaussian processes, allowing the model to scale to large data sets.

We frame the problem in relation to the mixture of experts model discussed in section 4.1. Gaussian processes can not be applied directly to the expectation maximisation learning algorithm used to train mixture of experts models. As we saw in section 4.1.1 each expert of a Bayesian mixture of experts model is trained using the weighted contributions of all the training points as in (4.8). This is equivalent to solving a regularized least squares problem where the residuals of each training point are weighted by the latent variables $z_{n,k}$. This likelihood formulation assumes that each training point is independently distributed. However a Gaussian process explicitly models the joint distribution of the input variables, as such the likelihood can't be factorized with respect to the individual training points. Further, we must ensure that the number of training points used to train each GP is small such that the computational costs are manageable.

To apply Gaussian processes in a mixture of experts setting we re-define the use of the latent variable $\mathbf{z}$ such that instead of representing probabilities, it represents an assignment. Thus $z_{n,k} = 1$ indicates that training point n is used to train expert k. In this section we look at how to set $\mathbf{z}$ such that each expert models a single mode of the predictive distribution, as well as how to train the logistic regression model to obtain predictive weights for each expert. In chapter 5 we introduce an algorithm for learning $\mathbf{z}$ automatically from the data similar to the EM algorithm.

It should be noted that the role of the latent variable in this model is very different to that of a GPLVM [44] or a GPDM [91]. In these models the latent variable represents a continuous low dimensional representation of the human's pose. In our model, the latent variables represent the discrete assignments of the training samples that are used

to train each expert. They play a role analogous to the latent variables in a Gaussian mixture model [8].

The predictive distribution for a test point $\mathbf{x}_*$ is given by

$$p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y}, \Theta) = \sum_{i=1}^{K} p(z|\mathbf{x}_*) p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{X}_{\vartheta_i}, \mathbf{Y}_{\vartheta_i}, \theta_i), \tag{4.24}$$

where $\Theta = \{\theta_i\}_{i=1}^{K}$, $\vartheta_i = \{z_{n,j} : \forall n \in N, j = i \wedge z_{n,j} = 1\}$ is a set of indices representing the training points used for each expert and θ_i are the expert's learnt hyper-parameters. As with a Bayesian mixture of experts, each expert contribution is weighted by the output of a logistical regression model given by $p(z|\mathbf{x}_*)$.

To find the training points for each expert, ϑ_i, we cluster the pose space using k-means to obtain a set of *expert centres*, points in the training set which will have an expert centred around them. We then find the nearest S points to each centre, setting $\mathbf{z}_{n,i} = 1$.

There are no constraints placed on $\mathbf{z}$ in this model, that is a single training point can belong to zero or more experts. Each expert has a fixed size S which we typically set to 100 points. We evaluate the effect of varying the expert size and the number of experts K in §4.4.1.

An individual Gaussian process is only able to give a prediction for one output variable, in this case one axis of a joint of the subject. We learn an individual GP for each output variable within an expert, this allows each output variable to behave differently with respect to the input features while still maintaining some structure between the outputs. Each expert contains a small range of globally coherent poses, thus even if one of the subjects joints is ambiguous, the GP for that joint will give a meaningful prediction based on the local training data of that expert.

4.3.1 Learning the Gating Function

To obtain a weight for each expert for a test point $\mathbf{x}_*$ we use a logistic regression model

$$p(z|\mathbf{x}_*) = \frac{e^{\mathbf{w}_i^T \mathbf{x}_*}}{\sum_{j=1}^{K} e^{\mathbf{w}_j^T \mathbf{x}_*}} \tag{4.25}$$

which gives the probability of the test input feature $\mathbf{x}_*$ belonging to expert i. The weights $\mathbf{W} = \{\mathbf{w}_i\}_{i=1}^{K}$ are learnt by maximising the penalised log likelihood

$$\hat{\mathbf{W}} = \underset{\mathbf{W}}{\operatorname{argmax}} [\log p(\mathbf{C}|\mathbf{W}, \mathbf{X}) + \log p(\mathbf{W})] \tag{4.26}$$

where $\mathbf{C}$ is an $N \times K$ matrix representing which expert each training point is assigned to. The first term $p(\mathbf{C}|\mathbf{W}, \mathbf{X})$ measures how well the predicted probabilities given in (4.4) match the correct expert assignments $\mathbf{C}$. The second term $p(\mathbf{W})$ is a regularisation prior to avoid over-fitting. We set $p(\mathbf{W}) = \exp(\lambda|\mathbf{W}|_1)$ where $|W|_1 = \sum_{i=1}^{K}\sum_{j=1}^{D}|w_{i,j}|$ is the l_1 norm. This prior has the effect of pushing irrelevant weights to zero allowing the model to select the relevant features for each class. The influence of the prior is controlled using the parameter λ which is set using cross validation.

Typically, the matrix $\mathbf{C}$ indicating the target classes is encoded as a 1-of-K binary matrix, where each training point is assigned to exactly one expert. However, in the context of this model, a single training point can potentially belong to zero or more experts, as such we evaluate a number of methods for assigning the expert assignments $\mathbf{C}$ in §4.4.3.

Previous work by Urtasun and Darrell [85] inferred test poses by constructing local Gaussian processes centred around the nearest neighbours of each test point. Each component of the predictive distribution corresponded to a GP centred around a different neighbouring test point, and the priors were set using the inverse predictive variance of each GP. Setting the priors in this manner causes the model to bias output modes that have a lower signal noise. This means that regions of the data set which have inherent ambiguity, such as fast movements containing motion blur in the image, will receive less importance in the prediction. This bias leads to a less accurate predictive distribution.

Dependence on selecting the nearest neighbours in the feature space makes the model sensitive to background noise. The online experts can be constructed around training images that have a similar background, as opposed to a similar pose. This leads to incorrect predictions by selecting the incorrect data and hyper-parameters for the prediction. We overcome this issue by learning a logistic regression model, which is able to select the relevant input features for expert selection. This gives us greater robustness to varying backgrounds and other sources of ambiguity in the images. Further, the online construction of each GP in [85] is computationally expensive, significantly slowing test inference. In §4.4.4 we compare our method to [85] method and show a consistent performance improvement.

Another benefit of our model is the intuitive predictive distribution. In our model, each component of the predictive distribution (4.24) corresponds to a separate mode of the output. However with the predictive distribution of the Urtasun and Darrell model, each mode can have multiple experts mapped onto it, and sometimes a mode can be missed entirely. We demonstrate this on a synthetic data set in figure 4.3. Urtasun and Darrell's model on the left hand side shows how they have multiple components

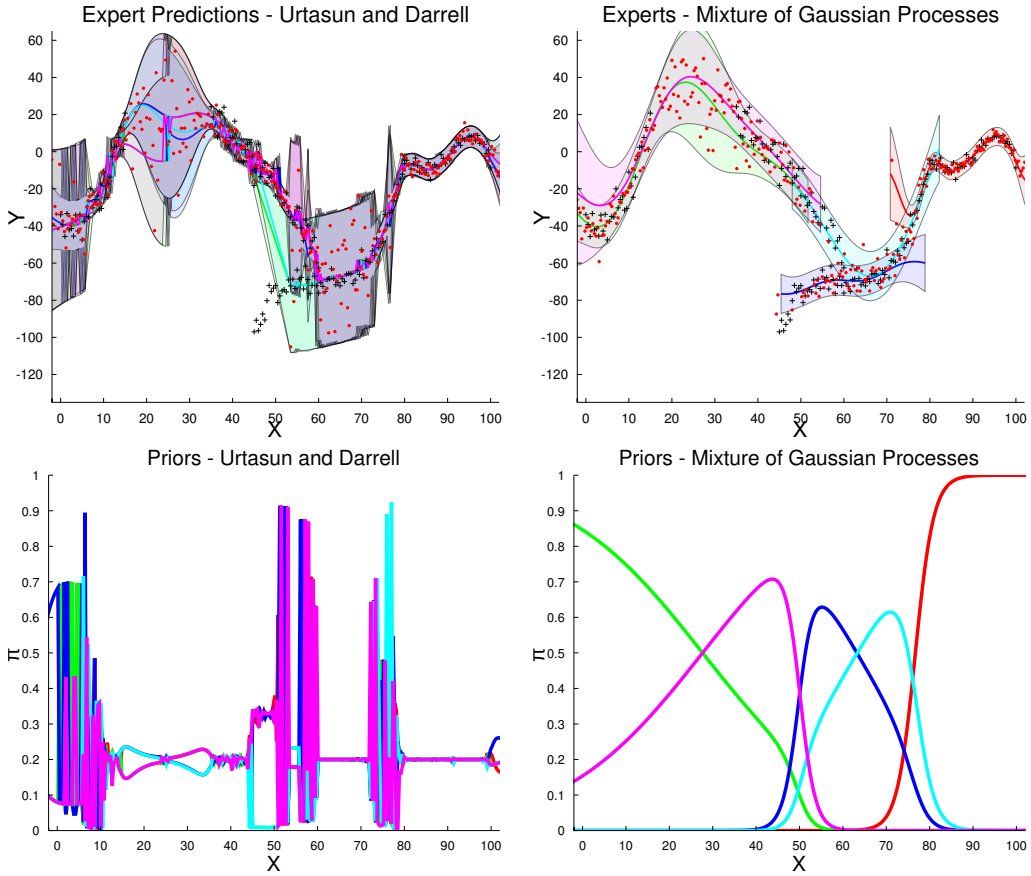

Figure 4.3: Comparison between the predictive distributions of the method in [85] and our proposed method. Upper plots show the expert predictions and the lower plots show the priors $p(z|\mathbf{x}_*)$. Training data shown as black crosses, each colour line and corresponding shaded region represent the predicted mean and variance of an expert. Red points are samples drawn from the predictive distribution. Both models are trained with 5 experts of size 50.

modelling the same predictive mode, and their model misses the multi-modal section. Our model on the right is able to model the multi-modal section, as well as give priors that can be interpreted as a weight for each expert. In chapter 5 we introduce a model that improves further on this by directly optimising $\mathbf{z}$.

4.4 EVALUATION

We evaluate our model on the ballet dataset [35], a sign language dataset [19] and HumanEva [74] as described in chapter 3. We evaluate the model using two types of image features, *bag-of-words* [2, 53, 49, 11] and HMAX [42, 85, 69]. The bag-of-words features are constructed following [49] using a codebook of 300 cluster centres. For ballet and HumanEva data set silhouettes are extracted and we use shape context descriptors [6]

extracted from the contour of the silhouette. Standard background subtraction techniques can't be used to obtain silhouettes on the sign language data set so we extract SIFT descriptors instead [47].

The ballet data set consists of a complex ballet choreography with 3D joint position annotations. The choreography is performed 5 times and we use 4 of the sequences for training and the final sequence for testing – resulting in 1601 training samples and 253 test samples.

The sign language data set is extracted from BBC television and consists of a continuous 6000 frame sequence. This is a very difficult sequence due to the moving background and image blur caused by fast movements. We break the sequences into 400 frame chunks and then randomly select chunks for the training and test sets to give 4400 training samples and 1200 test samples. We use 5 different random partitions of training and test data to evaluate the models.

The HumanEva data set consists of 3 subjects recorded from multiple cameras performing a range of actions. We evaluate our model by training a combined model for subjects S1, S2 and S3 using the images from a single camera (C3) for the Walking and Jog sequences.

Errors are given using the mean absolute error (MAE) measurement as described in section 3.1.2.

4.4.1 Expert Configuration

The main parameters of this algorithm are the number of experts K and the size of each expert S. A practical guideline for configuring the model is to choose a value for the expert size, typically $S = 100$, and then set the number of experts as $K = N/S$ where N is the number of training points. This way, each training point will belong to one expert on average.

Figure 4.4 demonstrates the effect of changing the number of experts K with a fixed expert size of 100. We set the number of experts using a multiplier α such that the number of experts is given by $K = \alpha N/S$. Thus setting α to 1 results in each training point belonging to one expert on average. Increasing α to higher values persuades a greater overlap between experts – giving greater certainty at boundary regions but increasing training time. In general, the performance tends to increase for more experts however saturates when α is set to 2.5. The performance starts to decrease slightly as lots of experts are added, this may because it makes learning the gating network more challenging.

Figure 4.5 demonstrates the effect of varying the expert size while holding the number of experts fixed. We train two models for each expert size, using a different number of experts for each. The number of experts are calculated by setting α to 1 for expert sizes 50 and 100.

The general pattern is that performance tends to increase with expert size up to a saturation point. While this varies with each sequence, 100–200 tends to be a suitable expert size. The ballet data set shows high sensitivity to the expert size, favouring each expert to have 100 points. Larger expert sizes cause a significant drop in performance, this may be because the ballet sequence is relatively short, approximately 300 frames, thus large expert sizes cause each expert to model a very wide range of poses in this short but complex sequence. The poor performance of models with 16 experts of size less than 75 could be attributed to the sparse coverage of the training data. As such, increasing the number of experts at this size leads to a significant performance increase.

4.4.2 *Expert Kernel Selection*

Gaussian processes optimise the kernel hyper parameters in order to gain a good fit of the training data. In this section we compare using an isotropic (ISO) kernel and a kernel with automatic relevance detection (ARD). Isotropic kernels have a single parameter p which is used to set the *lengthscale* of the kernel. Automatic relevance detection kernels learn an individual length scale for each input dimensionality [59]. This allows a Gaussian process with an ARD kernel to learn which dimensions of the input features are more relevant than the others.

Figure 4.6 shows a comparison between both kernels on each sequence with different features. As can be seen the ISO kernels out perform the ARD kernels in all experiments. This is a surprising result as one would expect that ARD kernels would identify the relevant features and give better test generalisation. We suspect that isotropic kernels perform better due to the fewer number of parameters that must be fitted. With an isotropic kernel there are 4 kernel parameters, as opposed to 304 parameters with an ARD kernel (see section 4.2).

4.4.3 *Training the Gating Network*

In this section we evaluate the most effective way of training the logistic regression model. This provides a weight over each expert's prediction and is important for ensuring that the model accurately represents the predictive distribution of the pose space.

We experiment with a number of ways for setting the target $N \times K$ probability

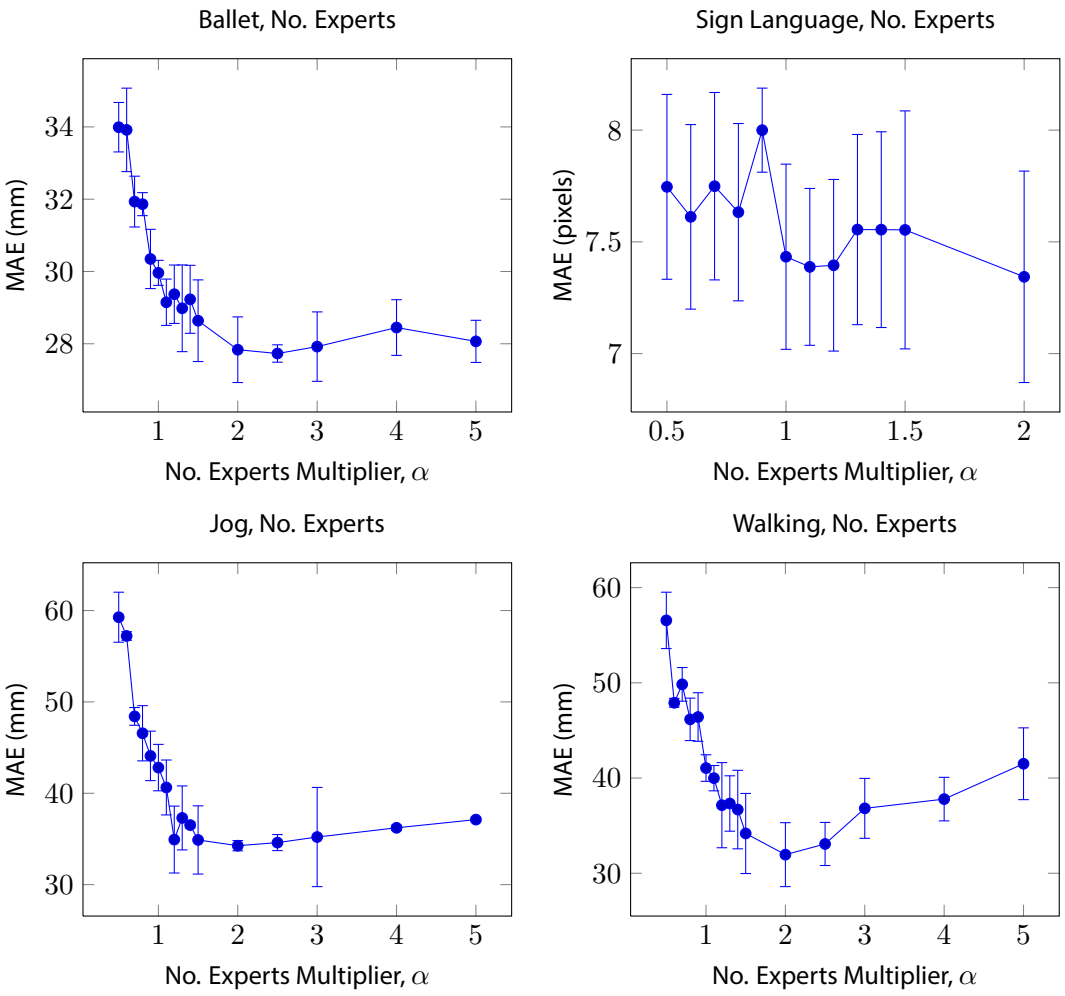

Figure 4.4: Evaluating different numbers of experts for each data set. The number of experts is given as a multiplier of the number of training examples N divided by expert size S. I.e. for multiplier x, the number of experts used for training is given by $x = N/S$. In these tests we use 100 points per expert. Errors are given in mean absolute error (section 3.1.2) with the results averaged over 5 runs. The standard deviation in the sign language results is comparatively higher because each run is performed over a different training and test partition.

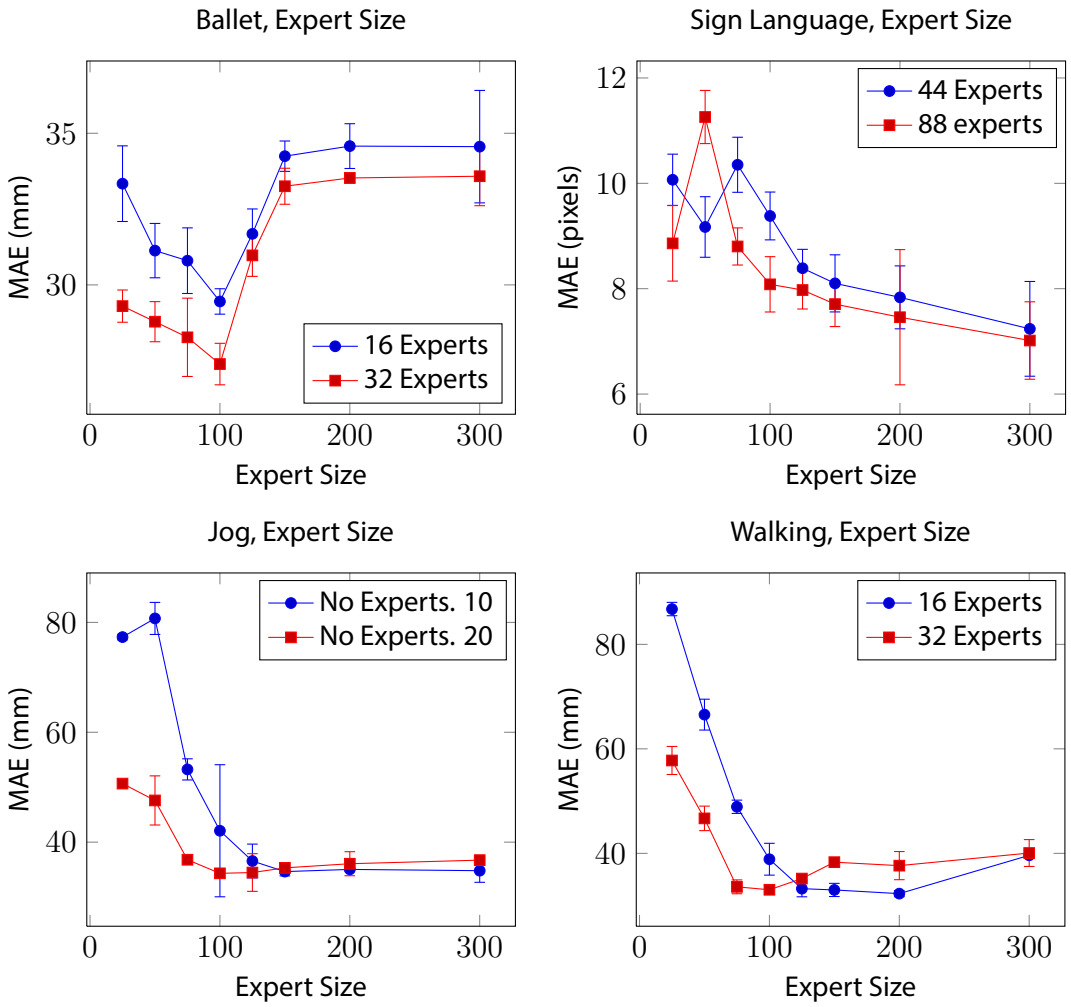

Figure 4.5: Demonstration of tracking errors in relation to the expert size. Errors are given in mean absolute error (section 3.1.2) with the results averaged over 5 runs. See text for discussion.

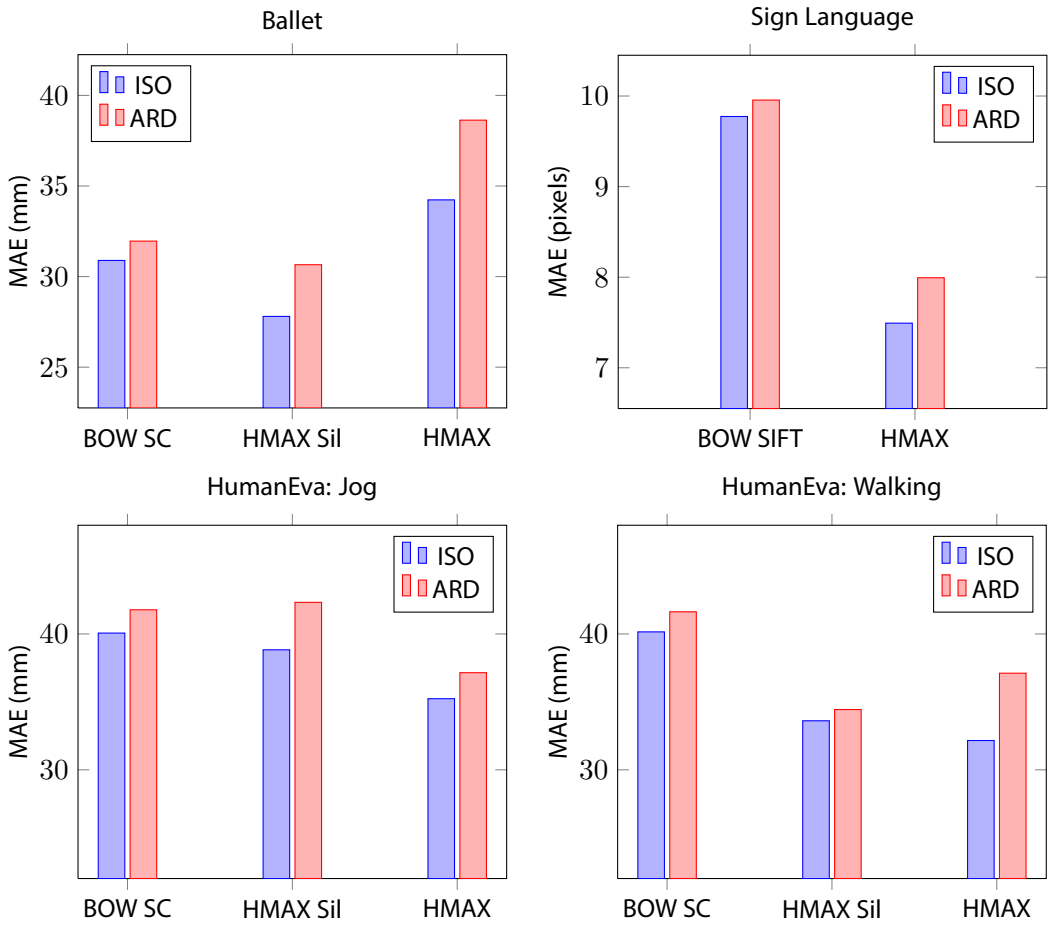

Figure 4.6: Comparison between ARD and ISO kernels.

matrix $\mathbf{C}$ from (4.26)

$$\textit{Expert Assignment} \quad \mathbf{C}_{n,i} = \begin{cases} 1 & \text{if } n \in \vartheta_i, \\ 0 & \text{otherwise.} \end{cases}$$

$$\textit{Nearest Expert Centre} \quad \mathbf{C}_n = \operatorname{argmin}_i |\mathbf{y}_n - \phi_i|,$$

$$\textit{Expert Density} \quad \mathbf{C}_{n,i} = p(\mathbf{y}_n | \mathbf{Y}_{\vartheta_i}).$$

Assignments of the form $\mathbf{C}_n = i$, where i is the expert index, are converted to a 1-of-K encoding. We also normalise the columns of $\mathbf{C}$ such that they sum to one and represent probabilities. The first approach directly uses the expert assignments to set $\mathbf{C}$. If a single training point has been assigned to two experts, then the normalisation step ensures that the target probabilities will assign it equal likelihood for each expert.

Setting $\mathbf{C}$ to the nearest expert centre assigns each training point to exactly one expert. We compute the distance of each training point to all the expert centres $\boldsymbol{\phi} = \{\phi_i\}$, and assign each point to its nearest centre. Finally, we fit a density model to the training poses to give a probability of each point belonging to each expert. The density models we experiment are a single Gaussian distribution and a Gaussian mixture distribution learnt using variational Bayesian inference [8].

Tables 4.1 and 4.2 compare the above methods on each data set. We train a fixed set of experts and compare how the different methods of setting the target probability matrix $\mathbf{C}$ affects the tracking accuracy. The expert assignment technique gives good performance all-round, giving a significant improvement on the ballet data set with HMAX features. On the sign language data set all methods give similar performance, with the nearest expert centre technique taking a small lead. The nearest expert centre technique performs poorly on the ballet data set with out background subtraction. This could be because the background noise causes ambiguities that need to be reflected in the priors. By using the nearest expert centre technique each training point in $\mathbf{C}$ is assigned to exactly one expert, thus $\mathbf{C}$ does not represent the ambiguity in the data.

In the HumanEva data set, the expert assignment technique often falls behind the others and nearest centre or the expert density techniques make better choices. The differences between these techniques are mostly minor, but the optimal technique should be chosen for each data set.

4.4.4 Comparison with Other Methods

We evaluate our mixture of Gaussian processes model (MGPR) against a selection of state of the art techniques. We compare to Bayesian mixture of experts BME [11], local

	Ballet		Sign Language
	HMAX Sil	HMAX	HMAX
Expert Assignment	**26.8** $\pm$ **0.39**	**30.9** $\pm$ **0.42**	7.3 $\pm$ 0.10
Nearest Expert Centre	27.6 $\pm$ 0.58	34.8 $\pm$ 0.82	7.5 $\pm$ 0.10
Expert Density Gaussian	**26.8** $\pm$ **0.45**	32.4 $\pm$ 0.56	**7.1** $\pm$ **0.09**
Expert Density GMM	28.7 $\pm$ 0.58	31.6 $\pm$ 0.52	7.8 $\pm$ 0.10

Table 4.1: Evaluating the most effective way of setting $\mathbf{C}$ for the ballet and sign language data sets. Errors are calculated for each frame using the MAE measurement given in section 3.1.2. The Ballet data set errors are given in millimetres and the sign language errors are given in pixels. These results give the mean and standard error over the entire test sequence.

	Jog		
	BOW SC	HMAX Sil	HMAX
Expert Assignment	45.3 $\pm$ 0.77	38.4 $\pm$ 0.70	38.6 $\pm$ 0.63
Nearest Expert Centre	44.5 $\pm$ 0.74	39.2 $\pm$ 0.87	**35.1** $\pm$ **0.50**
Expert Density Gaussian	**42.9** $\pm$ **0.72**	37.7 $\pm$ 0.79	38.1 $\pm$ 0.68
Expert Density GMM	44.6 $\pm$ 0.79	**37.4** $\pm$ **0.76**	38.3 $\pm$ 0.68
	Walking		
	BOW SC	HMAX Sil	HMAX
Expert Assignment	40.4 $\pm$ 0.85	39.6 $\pm$ 0.78	33.7 $\pm$ 0.56
Nearest Expert Centre	39.6 $\pm$ 0.83	**34.6** $\pm$ **0.74**	33.0 $\pm$ 0.55
Expert Density Gaussian	38.8 $\pm$ 0.81	38.0 $\pm$ 0.79	32.4 $\pm$ 0.54
Expert Density GMM	**38.1** $\pm$ **0.77**	37.6 $\pm$ 0.79	**32.3** $\pm$ **0.52**

Table 4.2: Evaluating the most effective way of setting $\mathbf{C}$ for the HumanEva data set. Errors are calculated for each frame using the MAE measurement in millimetres given in section 3.1.2. These results give the mean and standard error over the entire test sequence.

	Ballet			Sign Language	
	BOW SC	HMAX Sil	HMAX	BOW SIFT	HMAX
MGPR	32.5 ± 0.59	$\mathbf{28.1 \pm 0.49}$	32.9 ± 0.55	9.6 ± 0.03	7.3 ± 0.02
BME [11]	51.7 ± 0.93	62.0 ± 0.87	71.7 ± 0.84	12.9 ± 0.04	11.9 ± 0.03
Urtasun and Darrell [85]	36.1 ± 0.82	33.2 ± 0.79	38.3 ± 0.86	13.2 ± 0.03	8.1 ± 0.03
Random Forest	28.3 ± 0.42	31.4 ± 0.52	$\mathbf{31.4 \pm 0.47}$	$\mathbf{8.3 \pm 0.02}$	$\mathbf{7.0 \pm 0.02}$
sKIE [49]	$\mathbf{31.6 \pm 0.62}$	31.9 ± 0.67	37.6 ± 0.95	11.5 ± 0.07	9.0 ± 0.07
Kernel Regression	71.7 ± 0.85	71.7 ± 0.85	71.7 ± 0.85	12.1 ± 0.03	10.7 ± 0.02

Table 4.3: Quantitative results. Ballet results give the mean absolute error per joint represented as 3D joint positions in millimetres. Sign language results give the mean absolute error in 2D joint positions in pixels. Results are given along with their corresponding standard error. HMAX Sil and BOW SC features are extracted from silhouettes, HMAX and BOW SIFT are extracted directly from grey scale image.

shared kernel information embedding (SKIE) [49], Urtasun and Darrell's [85] local online Gaussian processes and *random forests* [17] and kernel regression [49]. Quantitative results are shown in tables 4.3 and 4.4 and visual results are shown in figure 4.9.

Our model performs favourably compared to the other techniques, offering a significant improvement of previous models based on using multiple Gaussian processes for regression [85]. This can particularly be seen in experiments which contain background noise. The model in [85] depends on using nearest neighbours in the image feature space to construct online experts – making the model sensitive to background image noise. This can be seen in the results for the BOW SIFT features on the sign language data set, and the HMAX features on the ballet data set. The BOW SIFT features on the sign language data set do a poor job of incorporating invariance to image information unrelated to the subjects pose, requiring a greater degree of noise invariance from the regression model. For the ballet data set, the HMAX features are computed without background subtraction, and it is on this experiment that we see the largest lead of our model compared to [85].

It should be noted that our technique is also able to perform very fast prediction due to it's offline-learnt models. Techniques such as SKIE and Urtasun and Darrell's local Gaussian processes build models online for each test point resulting in slow test inference.

Our model is often marginally out performed by random forest regression. Both techniques give similar performance and both support fast learning and prediction. The advantage of using our model is the compact mixture of Gaussian predictive distribution. This allows it to be directly incorporated into a dynamics framework such as [77].

Figure 4.7: Tracking results for the sign language dataset showing every fifth frame of a continuous sequence. Ground truth shown in red, predicted pose is shown in green.

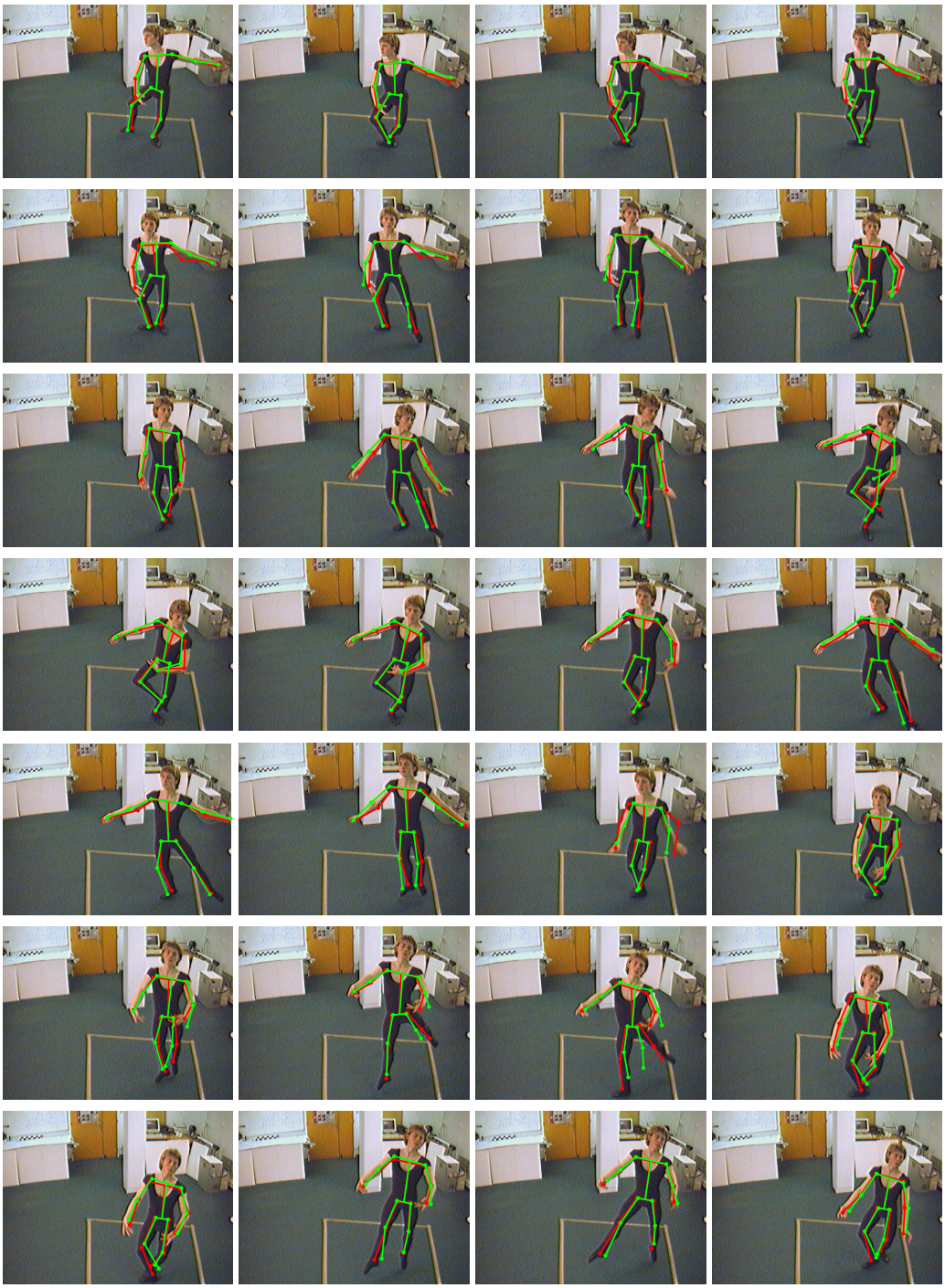

Figure 4.8: Tracking results for the ballet dataset showing every fifth frame of a continuous sequence. Ground truth shown in red, predicted pose is shown in green.

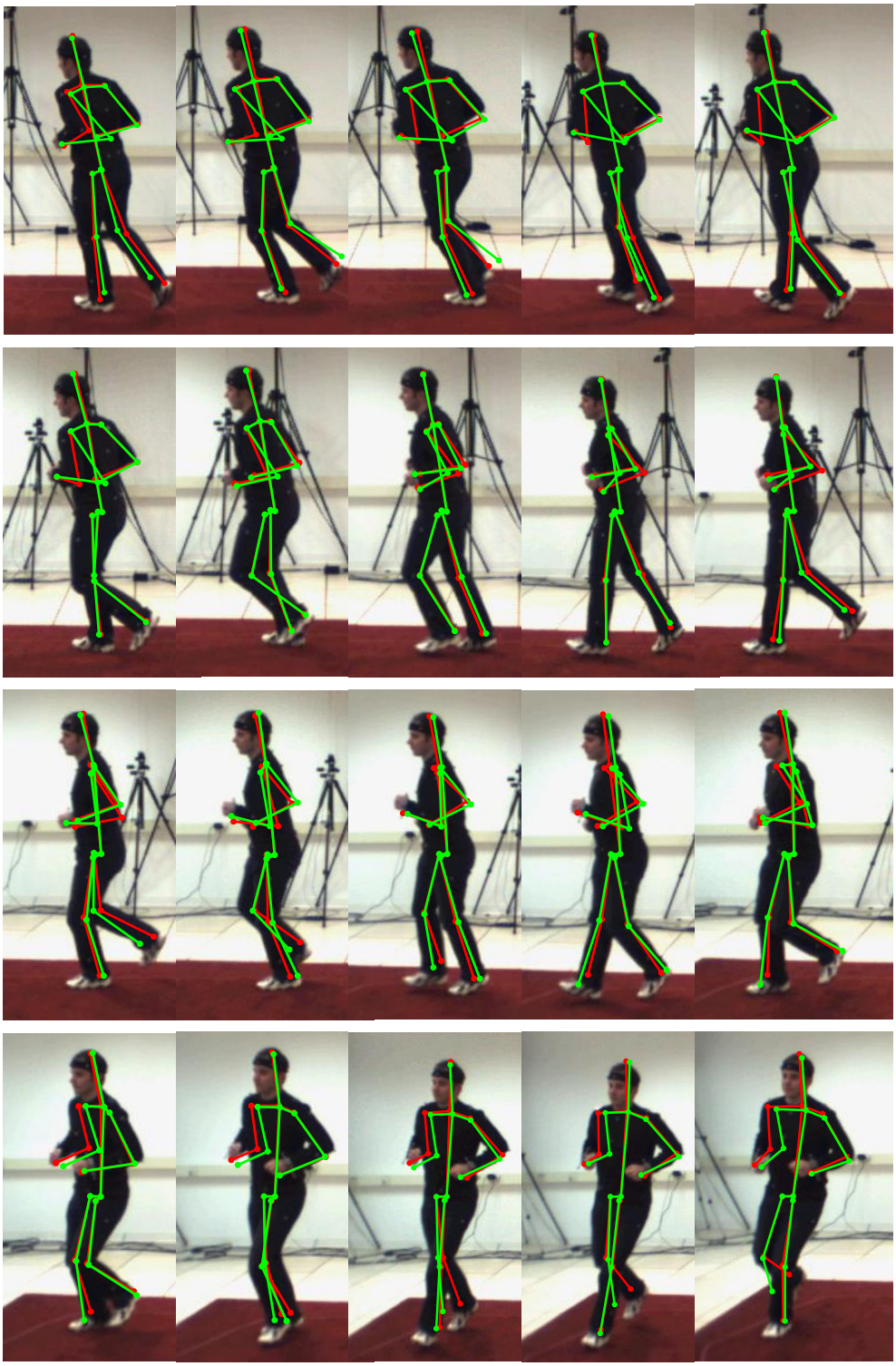

Figure 4.9: Tracking results for the HumanEva dataset showing every fifth frame of a continuous sequence. Ground truth shown in red, predicted pose is shown in green.

	Jog		
	BOW SC	HMAX Sil	HMAX
MGPR	40.1 ± 0.77	$\mathbf{35.4 \pm 0.62}$	34.5 ± 0.52
BME [11]	57.6 ± 0.91	81.0 ± 0.62	91.5 ± 0.44
Urtasun and Darrell [85]	43.7 ± 0.77	37.8 ± 0.66	37.2 ± 0.39
Random Forest	$\mathbf{39.8 \pm 0.44}$	39.7 ± 0.55	$\mathbf{34.1 \pm 0.37}$
sKIE [49]	40.0 ± 0.71	44.5 ± 0.84	41.4 ± 0.56
Kernel Regression	90.1 ± 0.24	90.0 ± 0.24	89.6 ± 0.24
	Walking		
	BOW SC	HMAX Sil	HMAX
MGPR	$\mathbf{40.5 \pm 0.66}$	$\mathbf{34.32 \pm 0.78}$	$\mathbf{32.09 \pm 0.45}$
BME [11]	56.7 ± 0.69	84.93 ± 0.34	86.02 ± 0.23
Urtasun and Darrell [85]	51.1 ± 0.85	45.00 ± 0.56	37.36 ± 0.37
Random Forest	46.0 ± 0.48	42.49 ± 0.46	32.87 ± 0.39
sKIE [49]	40.6 ± 0.86	38.13 ± 0.83	36.85 ± 0.57
Kernel Regression	86.5 ± 0.23	86.50 ± 0.23	86.02 ± 0.23

Table 4.4: HumanEva results given as mean absolute error in millimetres alongside the corresponding standard error. HMAX Sil and BOW SC features are extracted from silhouettes, HMAX and BOW SIFT are extracted directly from grey scale image.

4.5 DISCUSSION

In this chapter we have shown how the idea of a mixture of experts model can be applied to Gaussian processes. We started by reviewing mixture of experts models which combine a mixture of linear predictors to model non-linear and multi-modal mappings. Each linear predictor can be extended to model non-linear functions by encoding the input features using kernel basis functions increasing the accuracy of the learnt model. However these models have an ill-formed predictive variance that collapses towards zero when the test point moves away from the training data. This leads to confident but incorrect predictions. This problem is particularly prevalent when sparse predictors are used such as RVM and SVM which prune the available training data at test time.

We introduce a novel algorithm for using Gaussian processes in a mixture of experts framework. Using Gaussian processes in this way allows them to model large data sets with large amounts of ambiguity and multi-modalities. By training each Gaussian process expert on a local region of the training data, we can overcome the $O(N^3)$ training complexity – allowing efficient learning and prediction. We have shown that our model is able to give state of the art performance on human pose estimation data sets compared to other leading regression techniques.

5 Optimising Expert Locations

In this chapter we introduce a novel model for automatically optimising the size and location of the experts in a mixture of Gaussian processes model. As can be seen in the previous chapter, these models can be sensitive to the expert size and placement. We explore an algorithm that uses a Gibbs sampling approach to optimise the training points used to train each expert with respect to the model's predictive distribution. In §5.1 we cover relevant models in the machine learning literature and explain why these models cannot be directly applied to discriminative human pose estimation. In §5.2 we introduce our model which overcomes these limitations and in §5.3 we evaluate our proposed model and show its performance in comparison to the model discussed in the previous chapter. Parts of this work have been published in [28].

5.1 RELATED WORK

5.1.1 Infinite Mixtures of Gaussian Processes

Techniques which employ multiple Gaussian processes (GP) in a mixture model format have been used in the machine learning literature to model smaller problems which have similar properties to human pose estimation – multi-modality and varying ambiguity. Rasmussen and Ghahramani [61] introduce an infinite mixture of Gaussian processes model which uses a set of indicator variables to determine which expert each training point belongs to. The expert indicators $\mathbf{z} = \{z_n\}_{n=1}^{N}$ represent a set of discrete variables, where $z_n = i$ indicates that training point n belongs to expert i. The expert indicators are optimised using Gibbs sampling. Gibbs sampling is a *Markov chain monte-carlo* (MCMC) [8] technique for optimising a set of discrete variables. Each variable is sampled in turn from a proposal distribution conditioned on the remaining variables which are held fixed. By iteratively repeating this process the states of the individual variables converge to a local maximum.

A Dirichlet process [60] is used as a prior over the expert indicators allowing a possibly infinite number of GP experts. The Dirichlet process models the probability of an indicator variable z_n being assigned to an existing expert, $i \in \{1, \ldots, K\}$, or to a new expert $i = K + 1$. To express this distribution it is useful to introduce the *occupancy number* of an expert n_i, which gives the number of points which belong to expert i

$$n_i = \left| \left\{ z_j = i : j \in \{1, \ldots, N\} \right\} \right|. \tag{5.1}$$

The probability of an indicator variable z_n taking on an existing value $i \in \{1, \ldots, K\}$, or a new value $i = K + 1$ is given by

$$\forall i \in \{1, \ldots, K\}, \quad p(z_n = i | \mathbf{z}_{/n}, \alpha) = \frac{n_{i,/n}}{(N-1) + \alpha}, \tag{5.2}$$

$$i \in \{K + 1\}, \quad p(z_n = i | \mathbf{z}_{/n}, \alpha) = \frac{\alpha}{(N-1) + \alpha}, \tag{5.3}$$

where $n_{i,/n}$ gives the occupancy number of expert i with the n^{th} training point removed with $\mathbf{z}_{/n}$ taking on an analogous meaning. The probability of z_n being assigned to an expert i is proportional to the number of existing points in that expert. The probability that new experts are created is governed by the parameter α.

This gives a Dirichlet process a clustering property such that as the occupation number of an expert increases, the probability of points being assigned to that expert increases. The parameter α plays an important role in balancing the probability of a training point being assigned to an existing expert or to forming a new expert. The authors sample this parameter from a Gamma prior $\text{Gamma}(\alpha | a_\alpha, b_\alpha)$ [8] with fixed parameters a_α and b_α. By allowing this parameter to vary, it gives the model more flexibility in adapting the number of experts.

As discussed above, Gibbs sampling re-samples each indicator variable from a proposal distribution giving the probabilities of z_n taking on a value in $i \in \{1, \ldots, K+1\}$. For a training point n, the proposal distribution is given by

$$p(z_n = i | \mathbf{X}, \mathbf{Y}, \theta_i, \phi) \propto \underbrace{p(z_n = i | \mathbf{x}_n, \mathbf{z}_{/n}, \phi)}_{\text{Dirichlet gating}} \underbrace{p(\mathbf{y}_n | \mathbf{x}_n, \mathbf{X}_{\vartheta_i/n}, \mathbf{Y}_{\vartheta_i/n}, \theta_i)}_{\text{Expert likelihood}}, \tag{5.4}$$

which is the product of the Dirichlet gating function discussed above and the GP expert likelihood. ϑ_i/n is an index set which selects the training data associated with expert i excluding the n^{th} point $\vartheta_i/n = \{m : m \in \{1, \ldots, N\}, z_m = i, m \neq n\}$.

The expert likelihood term of (5.4) gives the likelihood of the training pair $(\mathbf{x}_n, \mathbf{y}_n)$

conditioned on the Gaussian predictive distribution of expert i

$$p(\mathbf{y}_n|\mathbf{x}_n, \mathbf{X}_{\vartheta_i/n}, \mathbf{Y}_{\vartheta_i/n}, \theta_i) = \mathcal{N}(\mathbf{y}_n|\mu_i(\mathbf{x}_n), \Sigma_i(\mathbf{x}_n)) \tag{5.5}$$

where $\mu_i(\mathbf{x}_n)$ and $\Sigma_i(\mathbf{y}_n)$ are given by a Gaussian process prediction made on the training data of expert i with the n^{th} training pair removed. Evaluating (5.5) for a new expert $i = K + 1$ requires giving the likelihood of a Gaussian process with no training data. For the squared exponential kernel as used in their paper [61], this is given by a sum of the kernel's variance parameters.

The expert likelihood term has the effect of ensuring that the training data of each expert can be well-represented by a Gaussian process. If the training pair $(\mathbf{x}_n, \mathbf{y}_n)$ has a poor likelihood with respect to expert i then the probability of it being used to create a new expert increases. In the case that a single expert models a multi-modal region of the pose space, the predictive likelihood (5.5) of the expert's training data will be comparitively low due to the large predictive variance caused by the expert averaging over two modes. In such cases, a new expert will be created using a single training point from one of the modes. When the remaining training data is re-sampled it will join the expert corresponding to the correct mode for each point.

The Dirichlet gating term, $p(\mathbf{z}_n = i|\mathbf{x}_n, \mathbf{z}_{/n}, \phi)$, in (5.5) is defined using a local input dependent estimate of the occupation number $n_{i,/n}$ which is calculated using a kernel model

$$n_{i,/n} = (n-1)\frac{\sum_{j=1}^{N} k_\phi(\mathbf{x}_n, \mathbf{x}_j)\delta_{z_j,i}}{\sum_{j=1}^{N} k_\phi(\mathbf{x}_n, \mathbf{x}_j)} \tag{5.6}$$

where $k_\phi(\mathbf{x}, \mathbf{x}')$ is a Gaussian kernel function with parameters ϕ which give an individual bandwidth for each input dimension and $\delta_{\cdot,\cdot}$ is the Kronecker delta function. This allows the model to learn the relevance of individual input features for the gating prior. This estimate of the occupation number is then used to compute the Dirichlet gating term in (5.4) using the Dirichlet process probabilities given by (5.3).

To fit the model to a data set consisting of N training pairs $(\mathbf{x}_n, \mathbf{y}_n)$, Rasmussen and Ghahramani initialise the expert indicators $\mathbf{z}$ to contain a single value, representing a single expert. The learning algorithm iterates between re-sampling the expert indicators $\mathbf{z}$ using the Gibbs sampling technique and optimising the parameters of the Gaussian process experts, θ_i, and the Gaussian kernel parameters, ϕ. MCMC techniques are used to optimise each of the model parameters in turn from their prior distributions. These parameters include the GP hyper-parameters $\Theta = \{\theta_i\}_{i=1}^{K}$, the gating kernel parameters ϕ and the Dirichlet process parameter α. When new experts are created, the expert parameters are sampled from the prior distributions over the expert parameters.

The predictive distribution is given as a mixture of Gaussian distributions conditioned on the test input $\mathbf{x}_*$

$$p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y}, \Theta, \phi) = \sum_{i=1}^{K} p(z|\mathbf{x}_*, \mathbf{z}, \phi) p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{X}_{\vartheta_i}, \mathbf{Y}_{\vartheta_i}, \theta_i). \tag{5.7}$$

which consists of each GP expert prediction weighted by the kernel model which is adapted from (5.6) to give the probability of the test point belonging to each expert

$$p(z|\mathbf{x}_*, \phi) = \frac{\sum_{j=1}^{N} k_\phi(\mathbf{x}_*, \mathbf{x}_j)\delta(z_j, i)}{\sum_{j=1}^{N} k_\phi(\mathbf{x}_*, \mathbf{x}_j)} \tag{5.8}$$

Rasmusssen and Ghahramani learn an individual bandwidth parameter for each input dimension of the kernel function $k_\phi()$. Learning a kernel model in this way is only feasible for low-dimensional inputs due to the large number of samples that must be evaluated. As such, it is not feasible to directly apply this model to pose estimation problems which have hundreds of input dimensions and thousands of training samples.

5.1.2 Alternative Infinite Model

Meeds and Osindero [48] extend the above technique to have a generative model over the inputs. They model the input distribution using a Gaussian mixture model (GMM) [8] to model the inputs for each GP expert. Figure 5.1 shows a graphical model which demonstrates the generative model placed over the input space. The $\mathbf{x}_n$ and z_n variables have swapped such that the inputs $\mathbf{x}_n$ are conditioned on the expert indicators, z_n. This allows them to obtain a generative distribution $p(\mathbf{x}|z)$, generating novel inputs.

They model the inputs using a GMM for each expert, where the probability of an input $\mathbf{x}$ belonging to expert i is given by

$$p(\mathbf{x}|z = i) = \sum_{l=1}^{L} \pi_l \mathcal{N}(\mathbf{x}|\mu_{i,l}, \Sigma_{i,l}). \tag{5.9}$$

Each GMM has L components and is given by mean and covariance parameters $\mu_{i,l}$ and $\Sigma_{i,l}$. The model is learnt in a similar fashion to Rasmussen and Ghahramani's model [61], iteratively optimising the model parameters using MCMC techniques and re-sampling the expert indicators using Gibbs sampling. The predictive distribution is similar to [61] except that the gating network uses the likelihood of the test input $\mathbf{x}_*$ given the GMM gating network.

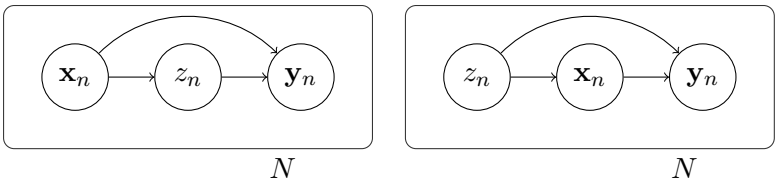

Figure 5.1: Illustration of infinite mixture of Gaussian process models. Left is Rasmussen and Ghahramani [61] and right is Meeds and Osindero [48]. In the latter model, the inputs $\mathbf{x}$ are conditioned on the expert indicators z to give a generative model over $\mathbf{x}$.

5.2 LOCAL EXPERT OPTIMISATION FOR HUMAN POSE ESTIMATION

These methods are not easily applied to human pose estimation problems which consist of large data sets with high dimensional features. The kernel classifiers used in [61] can be very sensitive to the bandwidth parameter. Learning these parameters for very large data sets can become expensive due to the large number of MCMC samples that must be evaluated to fit the high dimensional models. The GMM input model of [48] runs into statistical problems in human pose estimation data sets. Typically a single expert would contain 100 points and the input feature would contain 300D. Estimating a GMM to represent this data results in the components having low-rank covariance matrices – leading to inaccurate expert selection.

In this section we show how we can adapt our model from the previous chapter incorporating the ideas from the above models to automatically learn the size and locations of the experts. We maintain our logistic regression gating model and employ a multinomial distribution [8] over the expert indicators instead of the Dirichlet process used in [61, 48]. As such, our model deals with a finite number of experts. This does not pose a limitation for Human pose estimation tasks. These data sets contain large training sets necessitating a large number of small experts. This allows the $O(N^3)$ training complexity of each Gaussian process to be managed. By using a relatively small number of points in each expert, the training complexity is expressed as $O(KS^3)$ where K is the number of experts and S is their average size. By creating a large number of experts, such that S remains small, the training complexity scales linearly with the with the total size of the data set. As such, the ability to sample new experts is of less importance, as Human pose estimation problems require a large number of experts to be computationally feasible.

5.2.1 *Optimising the Expert Locations*

Gibbs sampling can be used to optimise the likelihood of a mixture of Gaussian processes model to ensure that each expert models an individual mode of the data set with coherent signal noise. To formulate this model we re-define the role of the latent variable $\mathbf{z}$ compared to the previous chapter. We assign each training point to exactly one expert by setting $z_n = i, i \in \{1, \ldots, K\}$.

We represent our predictive distribution as a mixture of Gaussian distributions with priors given by a gating model

$$p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y}, \Theta, \phi, \mathbf{z}) = \sum_{i=1}^{T} p(z|\mathbf{x}_*, \phi) p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{X}_{\vartheta_i}, \mathbf{Y}_{\vartheta_i}, \theta_i) \tag{5.10}$$

$$= \sum_{i=1}^{T} p(z|\mathbf{x}_*, \phi) \mathcal{N}(\mathbf{y}_*|\mu_i(\mathbf{x}_*), \sigma_i(\mathbf{x}_*)). \tag{5.11}$$

where $\mathbf{z} = \{z_n\}_{n=1}^N$, $z_n \in \{1 \ldots K\}$ indicate which expert each data point belongs to, $\vartheta_i = \{n|n \in N, z_n = i\}$ is the set of indices of data points that belong to expert i and $\Theta = \{\theta_i\}_{i=1}^K$. Each prediction is given by a Gaussian process $p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{X}_{\vartheta_i}, \mathbf{Y}_{\vartheta_i}, \theta_i)$ trained on a subset of the data ϑ_i and is weighted using a logistic regression model with parameters $\boldsymbol{\phi}$ that gives the probability of each expert conditioned on the input $p(z|\mathbf{x}_*, \phi)$.

5.2.2 *Learning the expert indicators* $\mathbf{z}$

The expert indicators, $\mathbf{z}$, control the size, location and number of experts and are set by Gibbs sampling over the predictive distribution. The probability of a data point n being assigned to expert i is given by

$$p(z_n = i|\mathbf{z}_{/n}, \mathbf{X}, \mathbf{Y}, \theta_i, \boldsymbol{\phi}) \propto p(\mathbf{y}_n|\mathbf{x}_n, \mathbf{X}_{\vartheta_{i/n}}, \mathbf{Y}_{\vartheta_{i/n}}, \theta_i) \\ p(z_n = i|\mathbf{z}_{/n}, \mathbf{x}_n, \boldsymbol{\phi}), \tag{5.12}$$

where $\vartheta_{i/n}$ is the index set ϑ_i with index n removed with an analogous meaning for $\mathbf{z}_{/n}$. To evaluate the probability of each point n belonging to each expert, we remove point n from the model and calculate it's likelihood with respect to each expert given the remaining training data. These likelihoods are then combined with the gating distribution given by the logistic regression model $p(z_n|\mathbf{X}, \phi)$ and normalised to form a multinomial distribution. The corresponding value of z_n is then set by sampling from the multinomial distribution formed from (5.12). This process is repeated for each point

in the data set, removing it from the model and re-sampling a value of z_n based on it's posterior likelihood with respect to each expert.

The Gibbs sampling step is performed iteratively, after a complete pass of the training data set, we update the expert and gating parameters Θ, ϕ, to represent the new expert locations. Algorithm 2 demonstrates this learning process. Training is initialised by setting the expert indicators $\mathbf{z}$ either randomly or by running K-means on the training pose data. The algorithm then proceeds in a similar fashion to expectation maximisation. In the expectation step we re-sample the expert indicators and in the maximisation step we update the Gaussian process hyper-parameters and the logistic regression weights. To detect convergence we calculate the log likelihood of the training data at each iteration. For a test input $\mathbf{x}_*$, a prediction is made using each expert as in (5.11) and the output is weighted by $p(z_* = i|\mathbf{x}_*, \phi)$.

When training the model with large data sets, the size of each expert has to be constrained to avoid individual experts growing such that they are computationally infeasible to train. This is achieved by modifying the distribution given in (5.12) such that the probability of a point being assigned to an expert is zero if it contains a chosen maximum number of points.

$$p(z_n = i|\mathbf{z}_{/n}, \mathbf{X}, \mathbf{Y}, \theta_i, \boldsymbol{\phi}) = \begin{cases} p(z_n = i|\ldots) & \text{if } n_i < S, \\ 0 & \text{if } n_i \geq S \end{cases} \tag{5.13}$$

where n_i is the number of points assigned to expert i.

The distribution $p(z_n = i|\mathbf{z}_{/n}, \mathbf{x}_n, \boldsymbol{\phi})$ gives the probability of the training input $\mathbf{x}_n$ belonging to expert i and is given by an L1 penalized multinomial logistic regression model [8]. The L1 penalty results in sparse weights allowing the model to select relevant input features.

5.2.3 *Comparison with Previous Methods*

Our proposed method differs from the previous models [61, 48] discussed above where Dirichlet process is used to create new experts during training. Instead we sample from a multinomial distribution given by $p(z_n = i|\mathbf{z}_{/n}, \mathbf{X}, \mathbf{Y}, \theta_i, \boldsymbol{\phi})$ (5.12). Models that place a Dirichlet process over the expert indicators rely on sampling a hyper parameter α from a Gamma distribution $\mathrm{Gamma}(\alpha|a_\alpha, b_\alpha)$ [8] which governs the probability of the model sampling a new expert. It is difficult to choose parameters a_α and b_α which lead to a suitable number of experts in pose estimation problems. We have observed that placing a Dirichlet prior over the expert indicators often results the model sampling far

too many new experts, resulting in a model with very few points in each expert. This results in very poor performance for pose estimation as the expert is not able to learn the image to pose mapping from so few training examples. Instead, our model is initialised with a large number of small experts, and unsupported experts are removed during the Gibbs sampling process.

The model is formulated in a multivariate setting, such that each expert represents a local set of full poses as opposed to optimising $\mathbf{z}$ individually for each output dimension. This has the advantage of imposing a degree of structure to the predictive distribution ensuring that predictions made are valid poses as observed from the training set.

Algorithm 2 Algorithm for learning mixture of GPs model.

for all $j \in \{1 \ldots$No. Gibbs iters $\}$ **do**
  for all $i \in T$ **do**
    Remove expert i where $\sum_{n=1}^{N} \delta(z_n, i) = 0$
  end for
  $\Theta \leftarrow \arg\max_{\boldsymbol{\Theta}} p(\mathbf{Y}, \mathbf{X}, \Theta, \mathbf{z})$
  $\phi \leftarrow \arg\max_{\phi} p(\mathbf{z}, \mathbf{X}, \phi)$
  for all $n \in \{1 \ldots N\}$ **do**
    $z_n \leftarrow \text{Multinomial}(\forall i \{p(z_n = i | \mathbf{z}_{/n}, \mathbf{x}_n, \alpha)\})$
  end for
end for

5.3 EVALUATION

We evaluate our proposed model on both synthetic data sets to check their validity and the pose estimation data sets used in chapter 4.

5.3.1 Demonstration on Synthetic Data

In this section we demonstrate the learning process on a synthetic data set. This data set allows us to visualise the expert locations and their predictive distribution as the learning algorithm updates the expert indicators $\mathbf{z}$. We use a data set from [48] consisting of 4 functions with varying levels of output noise and a multi-modal region. It comprises

of the following functions

$$\begin{aligned}
f_1(x_1) &= 0.25x_1^2 - 40 + \epsilon(\sigma_1) & x_1 &= (0, 15) & \sigma_1 &= 7 \\
f_2(x_2) &= -0.0625(x_2 - 18)^2 + 0.5x_2 + 20 + \epsilon(\sigma_2) & x_2 &= (35, 60) & \sigma_2 &= 7 \\
f_3(x_3) &= 0.008(x_3 - 60)^3 - 70 + \epsilon(\sigma_3) & x_3 &= (45, 80) & \sigma_3 &= 4 \\
f_4(x_4) &= -\sin(0.25x_4) - 6 + \epsilon(\sigma_4) & x_4 &= (80, 100) & \sigma_4 &= 2
\end{aligned} \tag{5.14}$$

where $\epsilon(\sigma) = \mathcal{N}(0, \sigma^2)$ gives a Gaussian noise process with standard deviation σ. The individual functions of the data set are shown in figure 5.2. This data set is ideal for demonstrating the model's ability to correctly identify the location of each expert. To obtain an accurate fit of this data set, each expert should be trained on a region with a coherent signal noise. For example, if a single expert was used to model f_3 and f_4, the predictive variance of the expert would average the signal noise of both functions, resulting in poor predictive uncertainty.

Figure 5.3 shows the learning process applied to the above data set. In these plots, the x-axis is analogous to the image features, and the y-axis is analogous to the pose. Thus this data set represents a 1D simulation of discriminative pose estimation, a multi-modal mapping with varying levels of ambiguity. The left hand column shows that the expert indicators $\mathbf{z}$ are initialised randomly, leading to a predictive distribution where each expert models a similar function. The middle column of figures shows the predictive distribution for each expert, and the right hand column shows the prior distribution $p(z|x)$.

Initially the predictive distribution is similar for each expert, the multi-modal region is averaged leading to very high signal noise with many inaccurately placed samples. To fit the model, we apply 50 iterations of algorithm 2 to the randomly initialised expert indicators. The second row shows the model after 5 iterations, the expert indicators have began to cluster on coherent portions of the data set that represent the underlying functions in (5.14). Particularly, the *red* expert is modelling f_4, and the *cyan* expert is modelling the portion of f_3 which corresponds to the lower of the two modes. The priors have also taken shape, reflecting the red expert's location on f_4. The predictive distribution now reflects the individual modes of the data set, however the cyan expert still generates some incorrect samples.

After further iterations of the Gibbs sampling algorithm, we see the cyan expert models more of f_3 and the green and blue experts model f_1 and f_2 respectively. The green expert models the whole of f_1 and part of f_2, this is a reasonable solution as both functions have identical signal noise and are therefore modelled well by a single Gaussian process. The predictive distribution is able to interpolate between the two

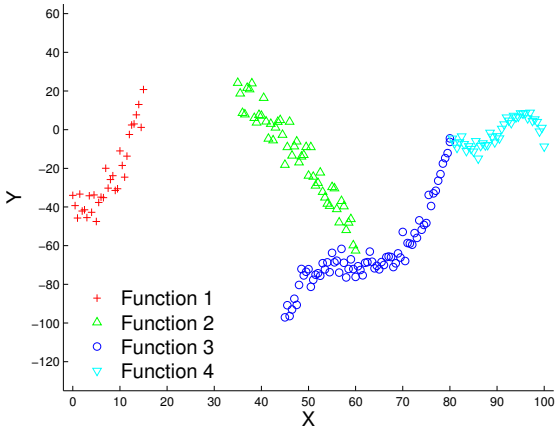

Figure 5.2: Synthetic data set for evaluating Gibbs sampling algorithm. See text for discussion.

functions, giving more uncertainty to this region where there is no training data. As the Gibbs iterations are completed, the priors given in the right hand column develop a more confident partitioning of the input space. The final solution gives an accurate fit of the data, the test samples drawn from the predictive distribution closely match the training data. The left hand plot of figure 5.4 shows the log likelihood of the training data through the Gibbs sampling process. The log likelihood converges after approximately 20 iterations and can be used to determine a sufficient number of Gibbs sampling iterations.

Figure 5.5 illustrates the effect of using K-means to initialise the expert indicators. The K-means initialisation of the expert indicators can give poor expert placement resulting in an erroneous predictive distribution. However after running the Gibbs sampling algorithm, the locations of the experts are updated leading to a more accurate predictive distribution.

5.3.2 *Evaluation on Pose Estimation Data Sets*

In this section we evaluate the model on the pose estimation data sets introduced in chapter 3. We use the same evaluation strategy in §4.4.4 comparing the model introduced in this chapter to other state of the art regression techniques including our own model from the previous chapter. We compare to Bayesian mixture of experts BME [11], local *shared kernel information embedding* (SKIE) [49], Urtasun and Darrell's [85] local online Gaussian processes, *random forests* [17] and kernel regression [49]. We train the Gibbs sampling model by initialising the expert indicators using K-means and then allowing 20 Gibbs sampling iterations to updates the expert locations. The right hand plot of figure 5.4 shows the log likelihood of the training data on the Ballet data set. After

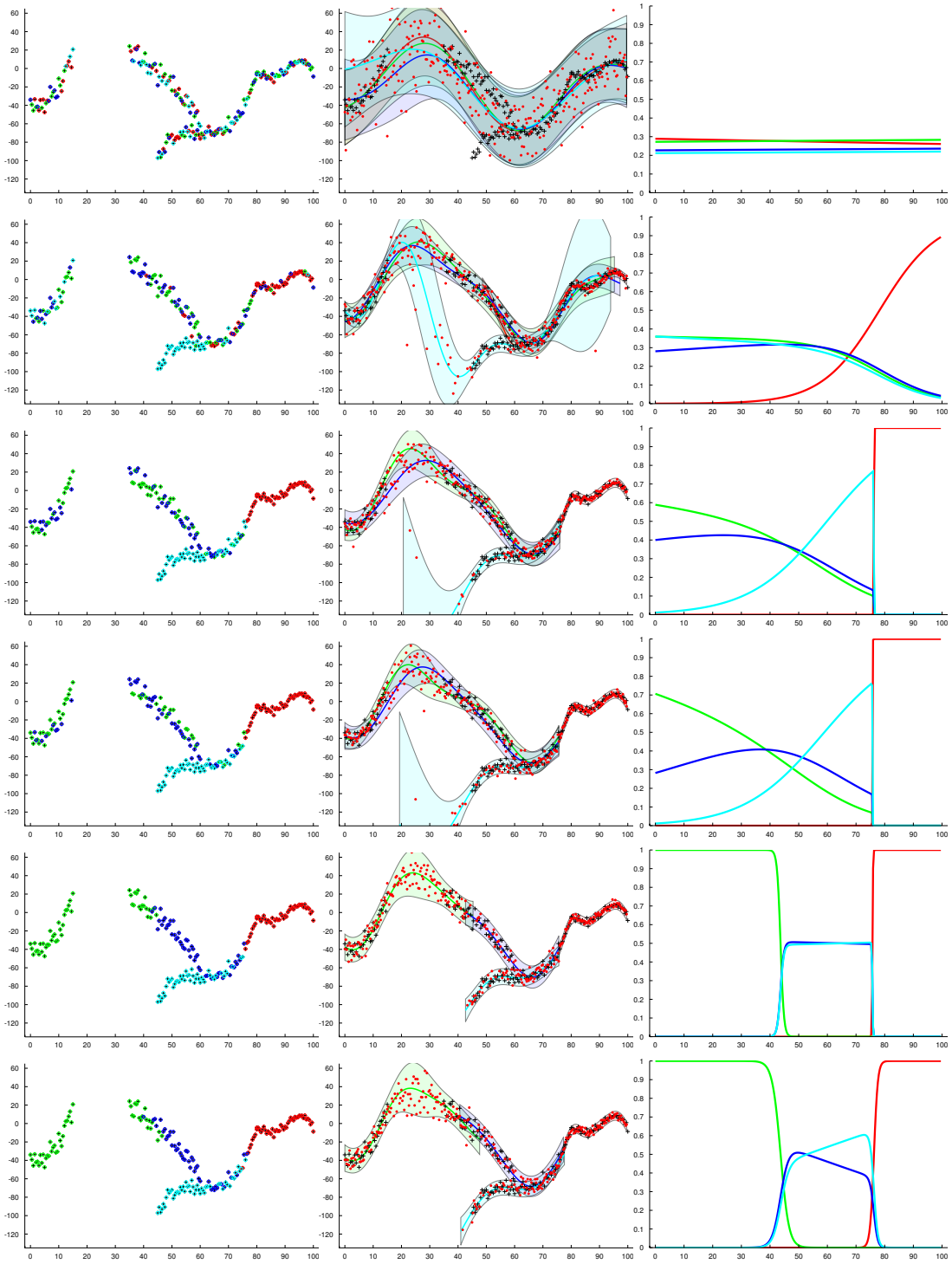

Figure 5.3: Mixture of Gaussian processes learning algorithm. Black crosses represent training data, each expert is represented by a different colour. Left, the expert assignments $\mathbf{z}$, middle, the predictive distribution, right, the expert priors, $p(z|\mathbf{x})$. From top to bottom shows the algorithm state after 0, 10, 20, 30, 40 and 50 Gibbs sampling iterations from a random initialisation.

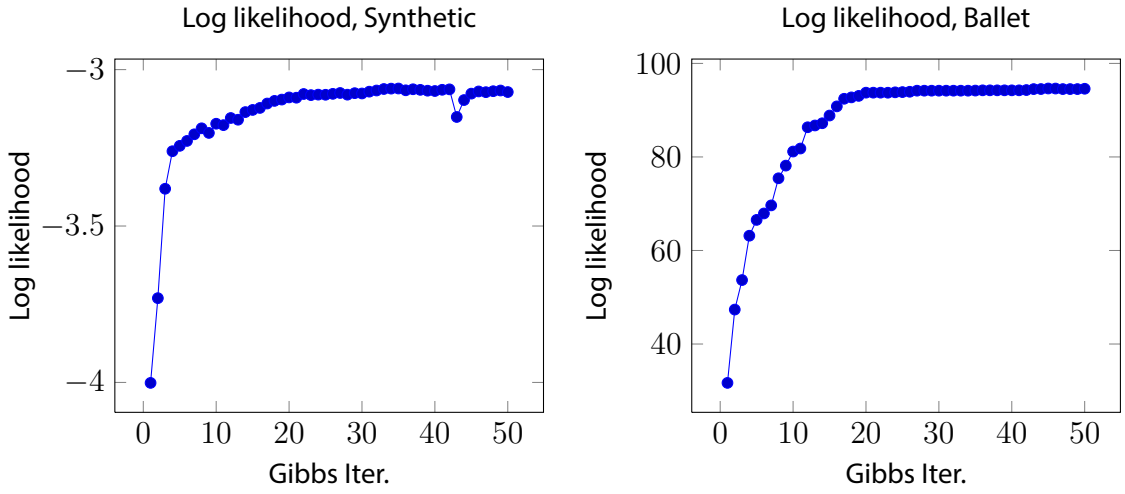

Figure 5.4: Plots showing the log likelihood of the training data at each gibbs iteration. Left hand plot shows the synthethic data set in (5.14) with the indicators initialised randomly as illustrated in figure 5.3. The Right hand plot shows the log likelihood on the Ballet data set, where the expert indicators are initialised using K-means.

20 iterations the log likelihood has flattened out, indicating convergence of the expert locations. Table 5.1 shows the pose estimation results achieved on the ballet and sign language data sets, and table 5.2 shows the HumanEva data set.

Overall, the Gibbs sampling model offers an improvement over the mixture of Gaussian processes model in previous chapters. Particularly the sign language data set, where the Gibbs sampling model gives the best performance out of all the techniques compared to. However we do see that on the Ballet data set, particularly with the silhouette HMAX features the previous model outperforms the Gibbs sampling model. The Gibbs sampling model gives strong performance on the HumanEva data set, often improving on sequences where the previous model struggles.

5.3.3 *Sensitivity to the Initial Number of Experts*

In this section we look at how varying the number of experts that the model is initialised with effects the resulting pose estimation accuracy. The formulation of this model, where each training point is assigned to exactly one expert, combined with the Gibbs sampling makes it less sensitive to the expert configuration compared to the previous model. In the previous model the number of experts reflected the coverage of the data set, there was no guarantee that each training point is used to train an expert. However here, the model formulation ensures that all regions of the data set are used in the training process, and the size of each expert is adapted individually to match the

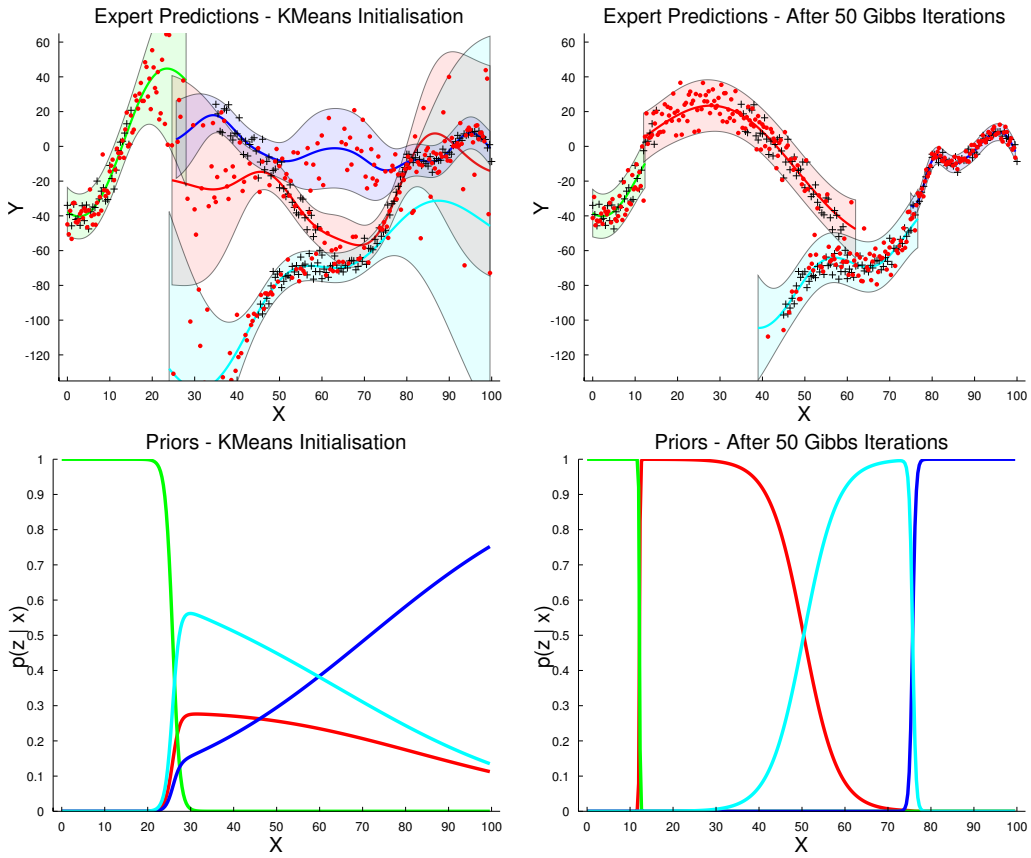

Figure 5.5: Best viewed in colour. Predictive distributions for the mixtures of Gaussian Processes model on the toy dataset from [61, 48]. The top row shows the expert predictions -- the black crosses represent the training points, the red dots are samples drawn from the predictive distribution and the coloured lines represent the predictive mean and variance of each expert. The bottom row shows the priors for each expert. See text for discussion.

	Ballet			Sign Language	
	BOW SC	HMAX Sil	HMAX	BOW SIFT	HMAX
MGPR Gibbs	32.5 ± 0.53	33.9 ± 0.58	33.1 ± 0.51	9.5 ± 0.03	$\mathbf{6.9 \pm 0.02}$
MGPR	32.5 ± 0.59	$\mathbf{28.1 \pm 0.49}$	32.9 ± 0.55	9.6 ± 0.03	7.3 ± 0.02
BME [11]	51.7 ± 0.93	62.0 ± 0.87	71.7 ± 0.84	12.9 ± 0.04	11.9 ± 0.03
Urtasun and Darrell [85]	36.1 ± 0.82	33.2 ± 0.79	38.3 ± 0.86	13.2 ± 0.03	8.1 ± 0.03
Random Forest	28.3 ± 0.42	31.4 ± 0.52	$\mathbf{31.4 \pm 0.47}$	$\mathbf{8.3 \pm 0.02}$	7.0 ± 0.02
sKIE [49]	$\mathbf{31.6 \pm 0.62}$	31.9 ± 0.67	37.6 ± 0.95	11.5 ± 0.07	9.0 ± 0.07
Kernel Regression	71.7 ± 0.85	71.7 ± 0.85	71.7 ± 0.85	12.1 ± 0.03	10.7 ± 0.02

Table 5.1: Quantitative results. Ballet results give the mean absolute error per joint represented as 3D joint positions in millimetres. Sign language results give the mean absolute error in 2D joint positions in pixels. Results are given along with their corresponding standard error. HMAX Sil and BOW SC features are extracted from silhouettes, HMAX and BOW SIFT are extracted directly from grey scale image.

	Jog		
	Walking		
	BOW SC	HMAX Sil	HMAX
MGPR Gibbs	**36.3 ± 0.59**	35.5 ± 0.57	**32.2 ± 0.41**
MGPR	40.1 ± 0.77	**35.4 ± 0.62**	34.5 ± 0.52
BME [11]	57.6 ± 0.91	81.0 ± 0.62	91.5 ± 0.44
Urtasun and Darrell [85]	43.7 ± 0.77	37.8 ± 0.66	37.2 ± 0.39
Random Forest	39.8 ± 0.44	39.7 ± 0.55	34.1 ± 0.37
sKIE [49]	40.0 ± 0.71	44.5 ± 0.84	41.4 ± 0.56
Kernel Regression	90.1 ± 0.24	90.0 ± 0.24	89.6 ± 0.24
	Walking		
	BOW SC	HMAX Sil	HMAX
MGPR Gibbs	**39.8 ± 0.58**	35.87 ± 0.63	32.41 ± 0.39
MGPR	40.5 ± 0.66	**34.32 ± 0.78**	**32.09 ± 0.45**
BME [11]	56.7 ± 0.69	84.93 ± 0.34	86.02 ± 0.23
Urtasun and Darrell [85]	51.1 ± 0.85	45.00 ± 0.56	37.36 ± 0.37
Random Forest	46.0 ± 0.48	42.49 ± 0.46	32.87 ± 0.39
sKIE [49]	40.6 ± 0.86	38.13 ± 0.83	36.85 ± 0.57
Kernel Regression	86.5 ± 0.23	86.50 ± 0.23	86.02 ± 0.23

Table 5.2: HumanEva results given as mean absolute error in millimetres alongside the corresponding standard error. HMAX Sil and BOW SC features are extracted from silhouettes, HMAX and BOW SIFT are extracted directly from grey scale image.

local data. Thus, creating a larger number of experts results in each expert containing less training points and vice-versa.

We evaluate how sensitive the model is to the number of initial experts for each data set. We express the number of experts using a multiplier α which expresses the number of experts as a function of the training data set size N. We give the number of experts as $K = \text{round}(\alpha N/100)$ such that setting α to 1 gives a model with each expert containing approximately 100 points. Thus, for the ballet data set with 1601 training frames, setting α to 1 results in a model with 16 experts.

Figure 5.6 shows the effect of varying the number of initial experts for each data set. The ballet data set performs better when the model is trained with more initial experts. This supports the results from §4.4.1 which showed smaller experts perform better on the ballet data set. This may be because the ballet data set consists of a repeated sequence of approximately 350 frames. If experts are too large then they represent a large portion of the sequence which may contain multi-modalities or varying ambiguity. Models with more initial experts are able to remove unnecessary experts during the Gibbs sampling process resulting in a better fit of the data.

The HumanEva and sign language data sets are less sensitive to this effect, but both

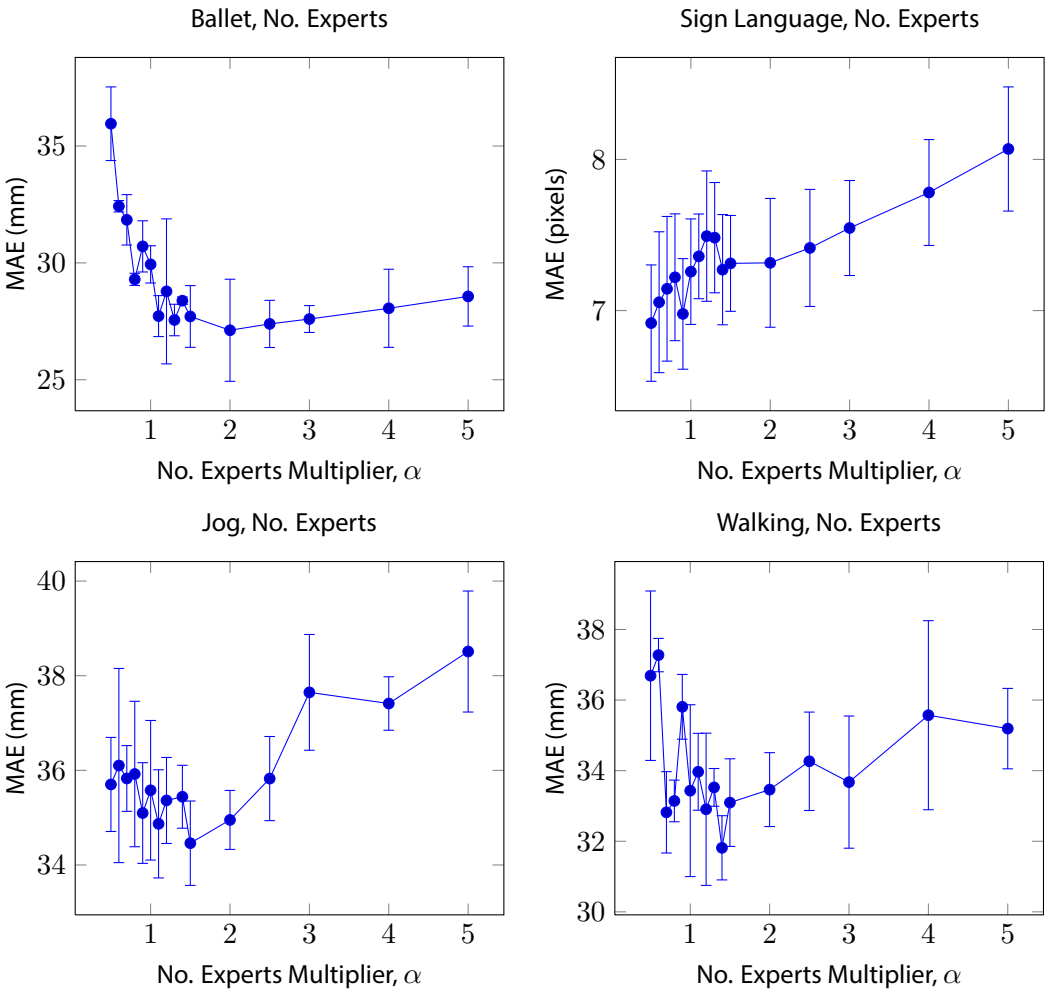

Figure 5.6: Evaluating the effect of varying the number of initial experts on pose estimation accuracy. Here we employ an expert multiplier α, where the number of experts K is given by $K = \alpha N/100$. Errors are given in mean absolute error (section 3.1.2) with the results averaged over 5 runs.

are shown to perform best when setting α close to 1. This results in each expert containing approximately 100 points, a result that is consistent with those given for the previous model.

5.4 DISCUSSION

In this chapter we have shown how Gibbs sampling can be used to optimise the size and locations of the experts in a mixture of Gaussian processes model. We reviewed the machine learning literature which uses infinite Gaussian process models for representing data sets with multi-modality and varying signal noises. The algorithms in the machine

learning literature rely on gating networks which are not scalable to human pose estimation problems. To model these data sets they require learning large numbers of parameters which is either computationally infeasible in the case of [61] or statistically unstable [48].

We introduce a novel algorithm for optimising the size and locations of the experts in a human pose estimation setting. Human pose estimation requires features that have hundreds of dimensions and data sets that have thousands of samples. Our model incorporates ideas from the methods discussed above but applies them in a manner which scales to these challenging data sets. This model allows each expert to model a region of pose that is uni-modal and has a coherent level of ambiguity.

Using a synthetic data set, we have shown that our model is able to correctly identify coherent portions of a data set that are well represented by a Gaussian process. We have also demonstrated that our proposed algorithm improves on our previous mixture of Gaussian processes model from chapter 4 when applied to human pose estimation tasks.

Dynamical Models for Discriminative Pose Estimation

6

Dynamical models play an important role in generative tracking where the role of the dynamics is to provide an initial estimate or a proposal distribution from which the optimal pose is obtained by evaluating pose samples against the image evidence. However when tracking using discriminative models, an accurate pose estimate is obtained directly through the mapping from image features to pose – there is no dependence on an initial pose estimate. As such, in discriminative tracking the role of a dynamical model is to incorporate temporal coherency into a sequence of pose estimates, and to use dynamical constraints to resolve multi-modal ambiguity in the image to pose mapping.

Discriminative pose estimation makes a estimate of a pose $\mathbf{y}$ as a function of an input $\mathbf{x}$ and some process noise ϵ

$$\mathbf{y} = f(\mathbf{x}) + \epsilon.$$

By estimating each frame of a sequence independently, the smoothness of human dynamics is replaced with a jittery signal due to the process noise ϵ and inaccuracies in the functional mapping $f()$. A dynamics model can be used to ensure that the pose predictions $\mathbf{y}_t$ for a sequence of frames $t \in \{1, \ldots, T\}$ are subject to a dynamical smoothness constraint.

This mapping from image to pose contains multi-modalities caused by ambiguous image evidence. This leads to the mixture of experts framework of previous chapters where the mapping is modelled using a multiple functional mappings, each modelling an individual mode of the ambiguous mapping. Dynamics can be used to disambiguate the image evidence by selecting correct mode of the appearance model.

Previous work on incorporating dynamics into discriminative models has been based

on conditioning the appearance model on a dynamics prediction [2]. A linear regression model is learnt to predict the pose from both appearance and dynamics expressed as the product of two kernels. The authors highlight the difficulty in learning the relative influence between the appearance and dynamics components in training their linear model.

Other researchers have applied dynamics models to the mixture of experts framework [83, 77]. Thayananthan et al. [83] learn a mixture of relevance vector machines (RVM) for their appearance to pose mapping. They integrate the prediction of each RVM into a particle filtering framework [38]. At each frame they combine their K RVM estimates with L estimates propagated from the previous frame to form $L \times K$ predictions. A generative image likelihood function is used to weight each of these predictions by comparing against the image evidence. The top L predictions are then propagated to the next frame using a linear dynamical system.

Sminchisescu et al. [77] learn a mixture of experts model which is conditioned on both the image features and the previous pose in a similar fashion to [2]. As with [83] they maintain L predictions at each frame. For a new image feature, they make a prediction using their mixture of experts model from each of the L previous poses to give L Gaussian mixture distributions, each with K components. A variational clustering approximation is used to obtain L predictions for each frame to propagate along the sequence.

In §6.1 we propose a method for incorporating a dynamics constraint which is specifically developed for models which give a mixture of Gaussian distributions as their predictive distribution. It uses a switching variable to consider each expert observation individually, and a dynamics programming algorithm to infer the optimal sequence of expert observations. This removes the necessity of the clustering approximation used at each frame in [77]. Learning is simplified as it doesn't rely on modifying the appearance model to be conditioned on a dynamics estimate which introduces difficult issues with balancing the influence of each regression input. Instead our model relies on the accurate Gaussian uncertainty of our mixture of Gaussian processes appearance model to balance the influence between appearance and dynamics. Parts of this work have been published in [28].

6.1 DYNAMICAL SECOND ORDER FILTERING FOR MIXTURE OF EXPERTS MODELS

In this section we introduce a novel algorithm that uses a dynamical model to infer a smooth path through a sequence of Gaussian mixture predictive distributions as given

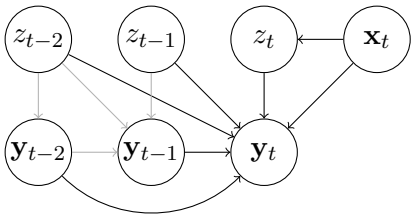

Figure 6.1: Graphical model for second order pose filtering showing the nodes involved in computing $\mathbf{y}_t$. See section 6.1.

by our mixture of Gaussian processes model. This algorithm differs from previous work [77] by using a latent variable to consider the individual components of the GMM instead of relying on a clustering approximation. The algorithm combines a multi-modal appearance model $p(\mathbf{y}_t|\mathbf{x}_t)$ and a separate dynamics model $p(\mathbf{y}_t|\mathbf{y}_{t-1}, \ldots)$ to infer a smooth path through a sequence of frames. This model allows dynamics and appearance terms to be learnt separately, avoiding the delicate problem of balancing the sensitivity between the two models [2].

A mixture of experts model gives a predictive distribution over the pose $\mathbf{y}$ as a mixture of Gaussian distributions as a function of an image feature $\mathbf{x}$:

$$p(\mathbf{y}|\mathbf{x}) = \sum_{i=1}^{K} p(z = i|\mathbf{x})\mathcal{N}(\mathbf{y}|\mu_i(\mathbf{x}), \Sigma_i(\mathbf{x})). \tag{6.1}$$

This is a general mixture of experts formulation, in this thesis we use our mixture of Gaussian processes model proposed in chapter 5.

In human pose tracking we wish to make a point estimate $\hat{\mathbf{y}}_t$ for the pose at frame t. The naive approach is to take the expectation of (6.1) by taking a weighted average of the component means. While this approach is acceptable in a uni-modal setting where only one of the Gaussian components is active, in multi-modal regions this averaging can result in incorrect poses. Secondly, the output of successive frames is often noisy due to the image ambiguities inherent in monocular pose estimation.

We propose an algorithm for inferring a smooth path through sequence of multi-modal pose estimations by formulating the problem as a second order Markov model. We introduce a latent variable which allows us to consider each expert of our appearance model individually, tracking multiple pose hypotheses through the sequence. The role of this latent variable is to act as a switch, separating out the components of the predictive distribution such that each forms a Gaussian appearance observation. By treating each appearance expert individually, we avoid the problem of an exponentially growing state conditional as when a GMM is propagated [8]. This simplifies inference

compared to previous methods [77] which propagate a GMM as their state distribution, enforcing the use of a clustering approximation to be made at each frame. The latent variable formulation allows us to derive a tractable algorithm for inferring the optimal sequence of latent states, giving us a pose estimate for each frame.

Our model is formulated such that we wish to infer two latent variables at each frame in the sequence, z_t denotes the expert representing a mode of the pose distribution, and $\hat{\mathbf{y}}_t$ represents a smoothed prediction within that mode. We maintain K predictions for each frame in the sequence, we denote the prediction at frame t from appearance expert i as $\hat{\mathbf{y}}_{i,n}$. The predicted pose at frame t by expert $z_t = i$ is given by:

$$\begin{aligned}\hat{\mathbf{y}}_{i,n} =& p(\mathbf{y}_t, z_t = i|\mathbf{x}_{1:t}) = p(\mathbf{y}_t|\mathbf{x}_t, z_t) \\ & \sum_{z_{t-2}} \sum_{z_{t-1}} p(\mathbf{y}_t|\hat{\mathbf{y}}_{z_{t-1},t-1}, \hat{\mathbf{y}}_{z_{t-2},t-2}) p(z_{t-1}|\mathbf{x}_{1:t-1}) p(z_{t-2}|\mathbf{x}_{1:t-2}) \end{aligned} \tag{6.2}$$

where $p(\mathbf{y}_t|\mathbf{x}_t, z_t)$ is the Gaussian expert prediction. $p(\mathbf{y}_t|\hat{\mathbf{y}}_{z_{t-1},t-1}, \hat{\mathbf{y}}_{z_{t-2},t-2)})$ is a dynamical prediction which uses the previous two states of $\hat{\mathbf{y}}$ to form a Gaussian prediction for $\hat{\mathbf{y}}_{z_t,n}$. Note the summation of z_{t-1} and z_{t-2}, the dynamical prediction is a weighted sum of the predictions from all combinations of previous locations. This reduces the sensitivity of the prediction to the previous pose estimates. Finally, $p(z_{t-1}|\mathbf{x}_{1:t-1})$ and $p(z_{t-2}|\mathbf{x}_{1:t-2})$ are the marginal probabilities of appearance mode being $z = i$ for a particular frame conditioned on the previous observations. These marginals represent the probability of an expert i making the appearance observation at frame t. We evaluate the marginals by integrating out the previous pose hypotheses

$$p(z_t|\mathbf{x}_{1:t}) = p(z_t|\mathbf{x}_t) p(\hat{\mathbf{y}}_{z_t,n}|\mathbf{Y}_{tr}) \sum_{z_{t-2}} \sum_{z_{t-1}} p(z_{t-1}|\mathbf{x}_{1:t-1}) p(z_{t-2}|\mathbf{x}_{1:t-2}), \tag{6.3}$$

where $p(\hat{\mathbf{y}}_{z_t,t}|\mathbf{Y}_{tr})$ is a density model which gives the probability of the posterior pose prediction $\hat{\mathbf{y}}_{z_t,t}$ being a valid pose. This has the effect of encoding the structure between the joints by down weighting pose predictions that aren't globally coherent with the training examples.

The prediction made for each expert at a given frame is a Gaussian product between the dynamical prediction $p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{y}_{t-2})$ and the appearance prediction $p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{z}_t)$. This has the useful property that the influence of each Gaussian distribution is inversely proportional to its uncertainty. That is, if an appearance expert has higher predictive variance, the dynamics model will have more influence over the prediction, and vice versa.

6.1.1 Modelling Human Dynamics

Human dynamics are relatively well modelled using a second order autoregressive process [2]. While this is not enough to model specific human actions or behaviours, it serves as a good general model of human motion. For the purposes of our dynamics framework, we require a dynamical model which generalises well to a high variety of poses as opposed to one that models specific actions.

We model $p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{y}_{t-2})$ using a linear regression model such that the prediction of the current pose is given by a linear projection of the two previous frames

$$p(y_{i,t}|y_{i,t-1}, y_{i,t-2}) = \mathcal{N}\big(\mu([y_{i,t-1}, y_{i,t-2}]^T), \sigma([y_{i,t-1}, y_{i,t-2}]^T)\big) \tag{6.4}$$

where $\mu()$ and $\sigma()$ are given by standard Bayesian linear regression [8]. We model the dynamics of each joint individually to simplify inference and allow for accurate modelling of the dynamics. Structure between each joint is enforced using a kernel density model $p(\hat{\mathbf{y}}_t|\mathbf{Y}_{tr}) = \sum_t^T k(\hat{\mathbf{y}}_t, \mathbf{y}_n)$ where $\hat{\mathbf{y}}_t$ is the predicted pose and $\mathbf{Y}_{tr} = \{\mathbf{y}_n\}_{n=1}^N$ are the training poses. Figure 6.2 shows a comparison between first and second order dynamics predictions. The first order dynamics in graph (a) contains a large amount of output ambiguity, for a single previous pose $\mathbf{y}_{t-1}$ there is a large range of possible current poses $\mathbf{y}_t$. Part (c) illustrates that a second order dynamics, $p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{y}_{t-2})$, lies on a linear manifold allowing accurate modelling using a second order linear regression model.

6.1.2 Inferring the Optimal States

In this section we show how we can use a dynamic programming algorithm for inferring an optimal sequence of states $\mathbf{z}_{1:T}$ and obtain a pose estimate for each frame. Inferring the optimum pose is performed by applying the max-sum algorithm [8] to the sequence. We initialise the algorithm by setting

$$p(\mathbf{y}_1|z_1, \mathbf{x}_1) = \boldsymbol{\mu}_{1,i}, \tag{6.5}$$
$$p(\mathbf{y}_2|z_2, \mathbf{x}_2) = \boldsymbol{\mu}_{2,i}, \tag{6.6}$$
$$p(z_1|\mathbf{x}_1) = p(z_1 = i|\mathbf{x}_1), \tag{6.7}$$
$$p(z_2|\mathbf{x}_2) = p(z_2 = i|\mathbf{x}_2), \tag{6.8}$$

where $\boldsymbol{\mu}_{t,i}$ is the predictive mean of component i given by the predictive distribution of appearance model and $p(z_1 = i|x_1)$ is the prior associated with each component (equation 6.1). We then proceed through the sequence evaluating $p(\mathbf{y}_t|z_t, \mathbf{x}_{1:t})$ for each

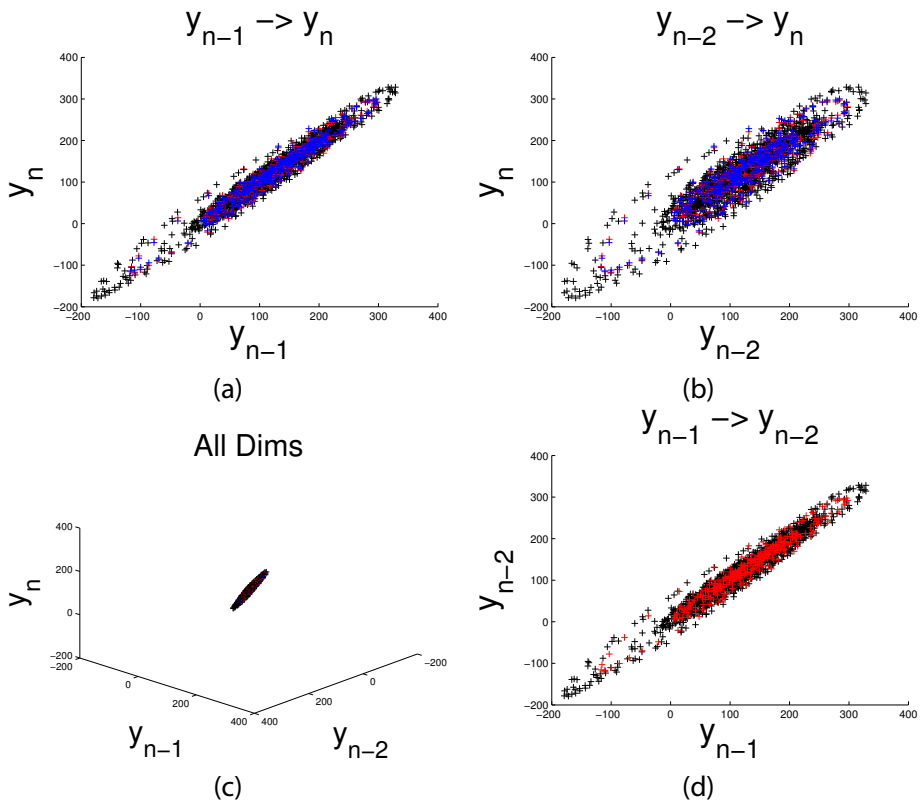

Figure 6.2: Best viewed in colour and zoomed. These plots show the pose data for the subject's left hand in one axis for the Ballet dataset. Black crosses are training points, red are test and blue are predicted from a linear model $p(y_t|y_{t-1}, y_{t-2})$. Plot (a) shows a first order prediction y_{t-1} against y_t, although there is clearly a linear relationship, there is a high degree of ambiguity. Plot (b) shows y_{t-2} against y_t and plot (c) shows a 3D plot with all three variables rotated to demonstrate the linear manifold. The second order pose distribution $p(y_t|y_{t-1}, y_{t-2})$ is highly linear, where y_{t-2} resolves vast majority of the ambiguity in plot (a). Plot (d) shows y_{t-1} plotted against y_{t-2}. The prediction of a linear model shown in blue is able to model the human motion to a high degree of accuracy.

appearance expert z_t and the corresponding marginal probabilities $p(z_t|\mathbf{x}_{1:t})$ (equations 6.2 and 6.3). At each frame, we store z_{t-1} and z_{t-2} which correspond to the most likely preceding appearance experts that lead to $z_t = i$:

$$\beta(i,n) = \underset{z_{t-1},z_{t-2}}{\operatorname{argmax}}\, p(z_t = i|\mathbf{x}_t)p(z_{t-1}|\mathbf{x}_{1:t-1})p(z_{t-2}|\mathbf{x}_{1:t-2}) \tag{6.9}$$

The second phase of the algorithm involves back-tracking through the stored sequences in β to extract the optimum sequence $\mathbf{z}$. We start by setting $z_T = \operatorname{argmax}_i p(z_T = i|x_{1:T})$ and $\hat{\mathbf{y}}_T = p(\mathbf{y}_T|z_T, \mathbf{x}_T)$, and then iterate backwards through the sequence $t = T-1, T-2, \ldots, 1$ setting:

$$z_t = \beta(z_{t+1}, t+1)_{z_{t-1}}, \tag{6.10}$$

$$z_{t-1} = \beta(z_{t+1}, t+1)_{z_{t-2}}, \tag{6.11}$$

$$\hat{\mathbf{y}}_t = p(\mathbf{y}_t|z_t, \mathbf{x}_{1:t}). \tag{6.12}$$

Thus $\hat{\mathbf{Y}} = \{\hat{\mathbf{y}}\}_{t=1}^T$ contains the smoothed predicted pose sequence and $\mathbf{z} = \{z_t\}_{t=1}^T$ stores the optimal sequence of experts that generated it.

6.2 EVALUATION

In this section we evaluate the dynamical pose filtering algorithm demonstrating its ability to smooth the predicted pose and resolve ambiguous frames. We compare our model against a standard linear dynamical system (LDS) whose parameters optimised the parameters using expectation maximisation [8]. An LDS infers a time-varying signal from a set of noisy observations by modelling it as a linear dynamical process with a linear-Gaussian observation distribution. Comparing against this model allows us to demonstrate the effectiveness of our algorithm at extracting a smooth pose signal compared to a baseline method. We manually tuned the dimensionality of the latent space to minimise the tracking errors while remaining numerically stable for the EM algorithm. For the HumanEva and sign language data sets we used a latent dimensionality that is half of the dimensionality of the full pose vector. This gave a good balance between smoothing and goodness-of-fit. For the ballet data set, the LDS was unable to fit the ground truth data with a large latent space. We use a 4 dimensional latent space, which is the largest that will fit the data without encountering statistical instabilities while learning.

We train the LDS on the ground truth pose data, and then use the Kalman smoothing algorithm [8] to obtain the predicted pose sequence by smoothing the expectation of

the appearance model's predictive distribution $\mathbb{E}[p(\mathbf{y}|\mathbf{x})]$ (6.1).

Figures 6.3, 6.4 and 6.5 show example frames for each data set where the dynamics model corrects the prediction of the appearance model. For the ballet data set these corrections typically correct a misplaced leg where the appearance model predicts as being bent for an individual frame. It also corrects some transitional poses between distinct frames where the appearance features are very subtle between close poses. The sign language data set is corrected when the appearance model makes incorrect predictions for frames where the subject is static, but background noise may cause an incorrect prediction. This dynamical prediction constrains the model from making large jumps which do not obey the learnt dynamical constraint. For the HumanEva data set, the dynamics model plays a key part in correcting rotations – where the predicted skeleton has the correct pose but is globally rotated incorrectly. The dynamics constraint ensures that the rotation of the subject does not jump around erratically when the image information is ambiguous.

Figures 6.6, 6.7, 6.8 and 6.9 shows example output of our dynamical pose filtering algorithm compared to the unfiltered expectation of the appearance model's predictive distribution, $\mathbb{E}[p(\mathbf{y}|\mathbf{x})]$, and a linear dynamical system. For the Ballet and HumanEva data sets we show the x, y, and z coordinates of the left hand and left foot. For the sign language data set we show the x and y coordinates of the right elbow, wrist and tip of the hand. These figures show that the dynamics model is able to smooth the prediction of the appearance model reducing the jitter between frames. Depending on a specific joint the linear dynamical system either predicts the same signal as the appearance model, or dramatically under fits the signal, resulting in poor tracking results.

Both the dynamical pose filtering algorithm and the linear dynamical system fail to track the HumanEva walking sequence in figure 6.9. They may be a limitation of the linear assumption of the dynamical propagation used in both models. A walking sequence such as this can contain significant non-linearities, particularly where the subject's feet are planted on the ground, a rapid deceleration in the joint's movement. Figure 6.10 shows the mean absolute error of each joint for our dynamical pose filtering algorithm applied to the walking sequence. Particularly high errors are observed on the subject's ankles which move with a highly non-linear motion.

Table 6.1 shows the effect on the mean absolute error on each data set. The dynamical pose filtering does not give a significant improvement in the overall tracking accuracy, giving slightly poorer results on the sign language data set. Although the tracking error is not improved significantly, the results are visually smoother with less jittery tracking than the appearance model alone. This shows a limitation with using the

Dataset	DPF	LDS	Appearance Only
Sign Language	7.1 ± 0.03	7.3 ± 0.04	$\mathbf{6.9 \pm 0.02}$
Ballet	$\mathbf{32.2 \pm 0.61}$	56.8 ± 1.76	33.9 ± 0.58
Jog	$\mathbf{31.8 \pm 0.38}$	34.9 ± 0.42	32.2 ± 0.41
Walking	$\mathbf{31.5 \pm 0.40}$	33.4 ± 0.36	32.4 ± 0.39

Table 6.1: Effect of dynamical pose filtering algorithm (DPF) on overall tracking errors compared to a linear dynamical system (LDS) and the appearance model alone.

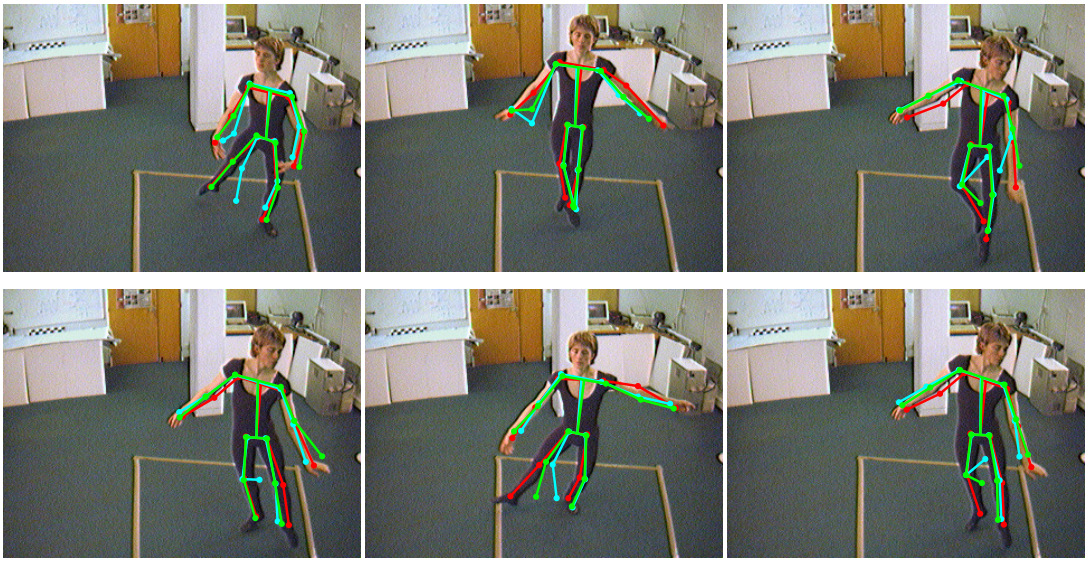

Figure 6.3: Example frames from the ballet data set where the dynamical pose filtering algorithm (green) is able to correct the appearance model (cyan). The ground truth pose is shown in red.

mean absolute error to evaluate human pose tracking algorithms. This measurement is unable to reflect temporal coherence of a pose sequence.

To quantify the smoothing effect of the dynamics model, we plot histograms of the disparities between consecutive frames in the predicted pose sequences. For each frame in a sequence we calculate the difference between the current frame and the previous frame, a measure of how much the subject's skeleton has changed in between two frames. For smoother tracking, the disparity between two consecutive frames should be smaller – leading to a histogram with a higher frequency of disparities in lower bins. In figures 6.11 and 6.12 we compare the disparities of consecutive frames between the predictions made by the appearance model alone, and the predictions made by the dynamic pose filtering algorithm. These histograms show that the dynamics model results in lower disparities on all four data sets, showing that it is able to smooth the pose predictions effectively.

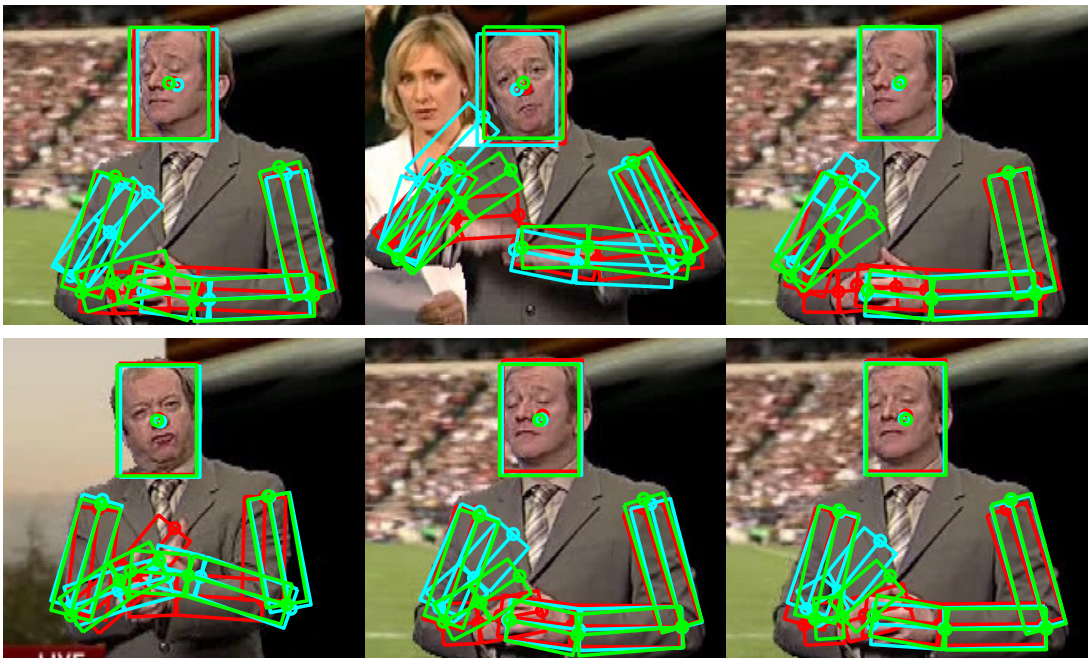

Figure 6.4: Example frames from the sign language data set where the dynamical pose filtering algorithm (green) is able to correct the appearance model (cyan). The ground truth pose is shown in red.

6.3 DISCUSSION

In this section we have introduced a novel algorithm for combining our mixture of experts model with a dynamical constraint in order to extract a smooth pose sequence from consecutive pose predictions. While the algorithm that we introduce is developed in the context of our mixture of Gaussian processes algorithm, it can be directly applied to any model with a Gaussian mixture predictive distribution. We show how by introducing a switching variable in our dynamic programming algorithm, we are able to remove the clustering approximation at each frame required by previous dynamical frameworks for Gaussian mixture models [77]. We demonstrate that our model is able to smooth the predicted pose sequence and correct some pose predictions where the appearance model has estimated the pose incorrectly.

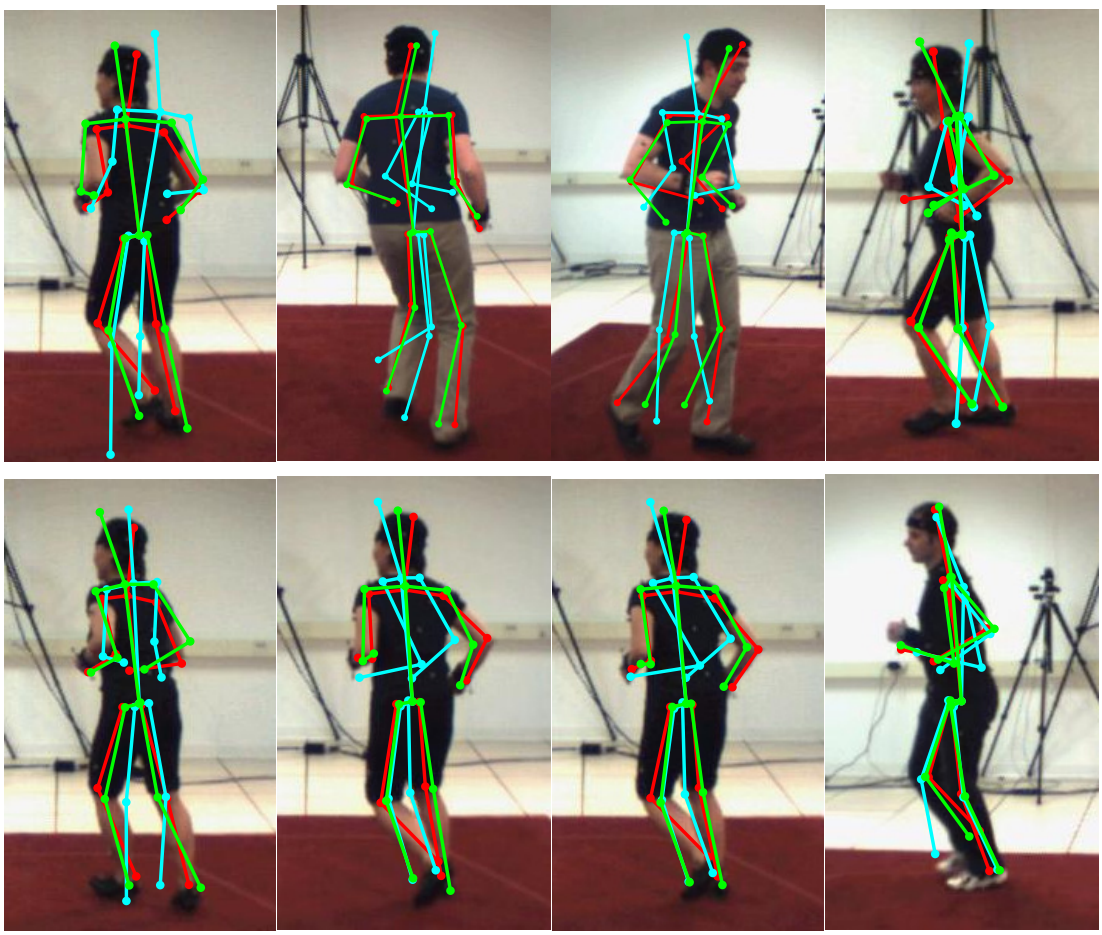

Figure 6.5: Example frames from the HumanEva data set where the dynamical pose filtering algorithm (green) is able to correct the appearance model (cyan). The ground truth pose is shown in red.

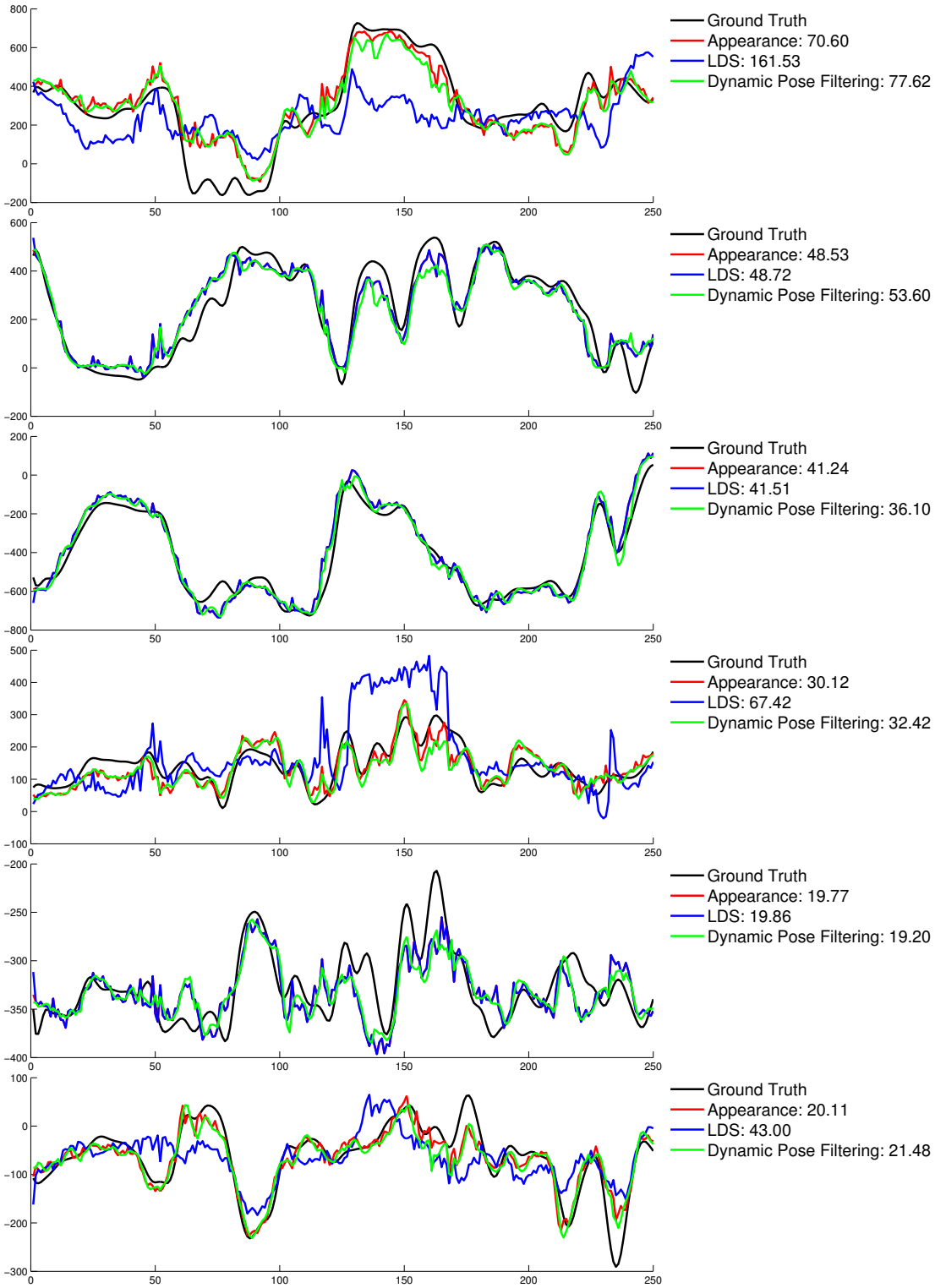

Figure 6.6: Joint position over time for the X, Y and Z axes of the left foot and left arm of the ballet data set. The black shows the ground truth, the red shows the appearance model, green is the dynamical pose filtering algorithm and blue is a linear dynamical system. Legend gives the mean absolute error for each model over this segment.

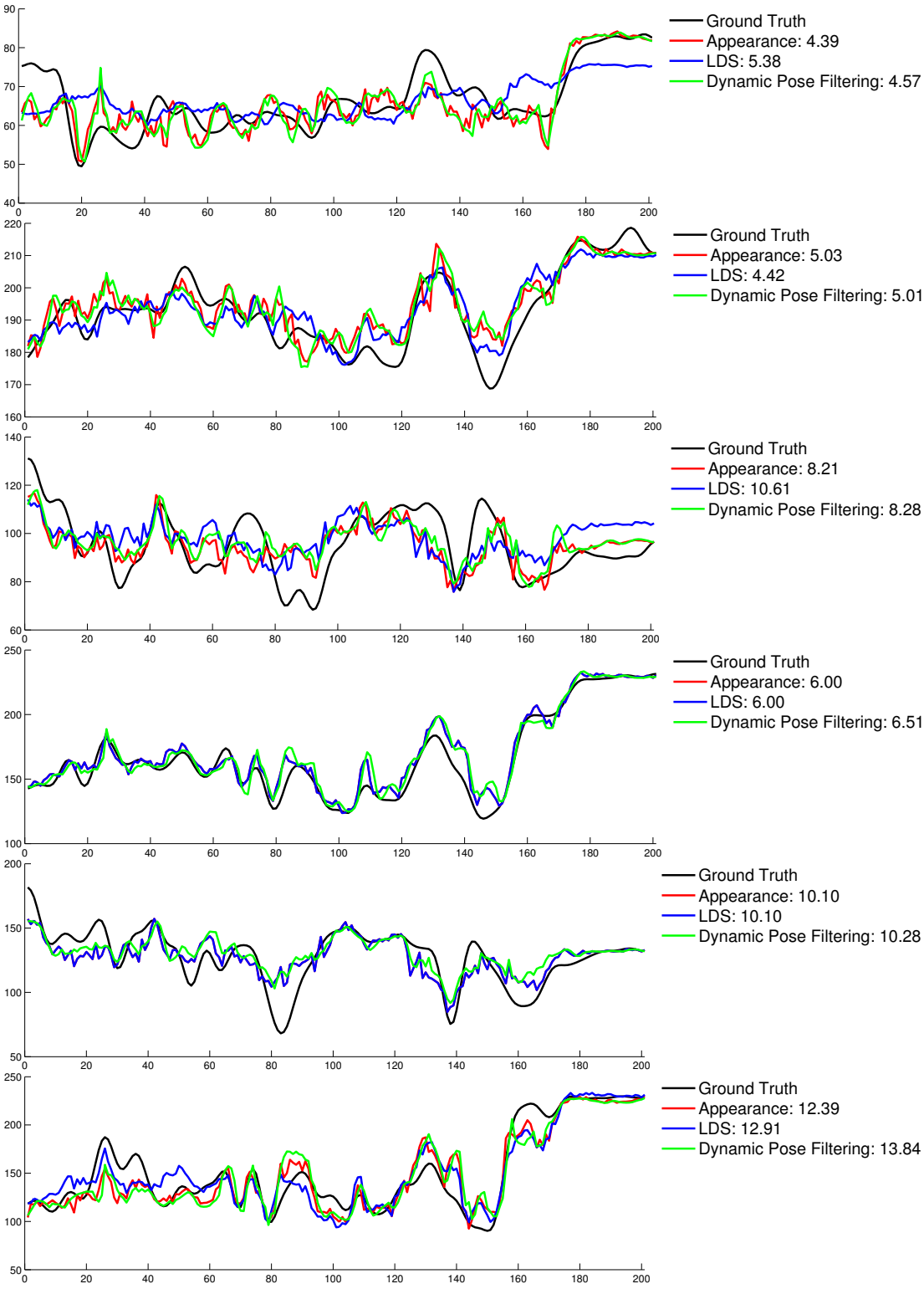

Figure 6.7: Joint position over time for the X and Y axes of the right elbow, wrist and tip of hand on the sign language data set. The black shows the ground truth, the red shows the appearance model, green is the dynamical pose filtering algorithm and blue is a linear dynamical system. Legend gives the mean absolute error for each model over this segment.

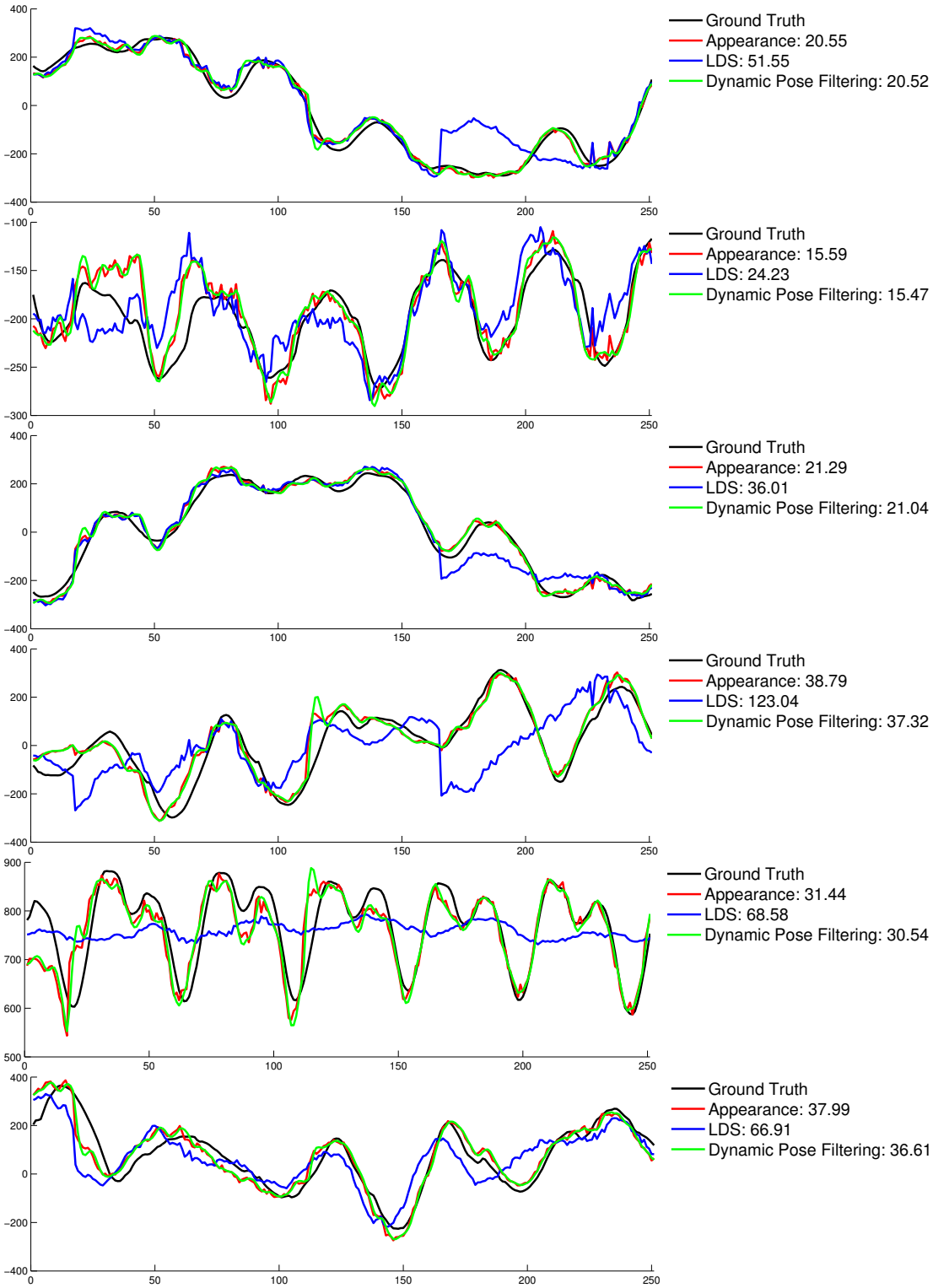

Figure 6.8: Joint position over time for the X, Y and Z axes of the left foot and left arm of the HumanEva Jog sequence. The black shows the ground truth, the red shows the appearance model, green is the dynamical pose filtering algorithm and blue is a linear dynamical system. Legend gives the mean absolute error for each model over this segment.

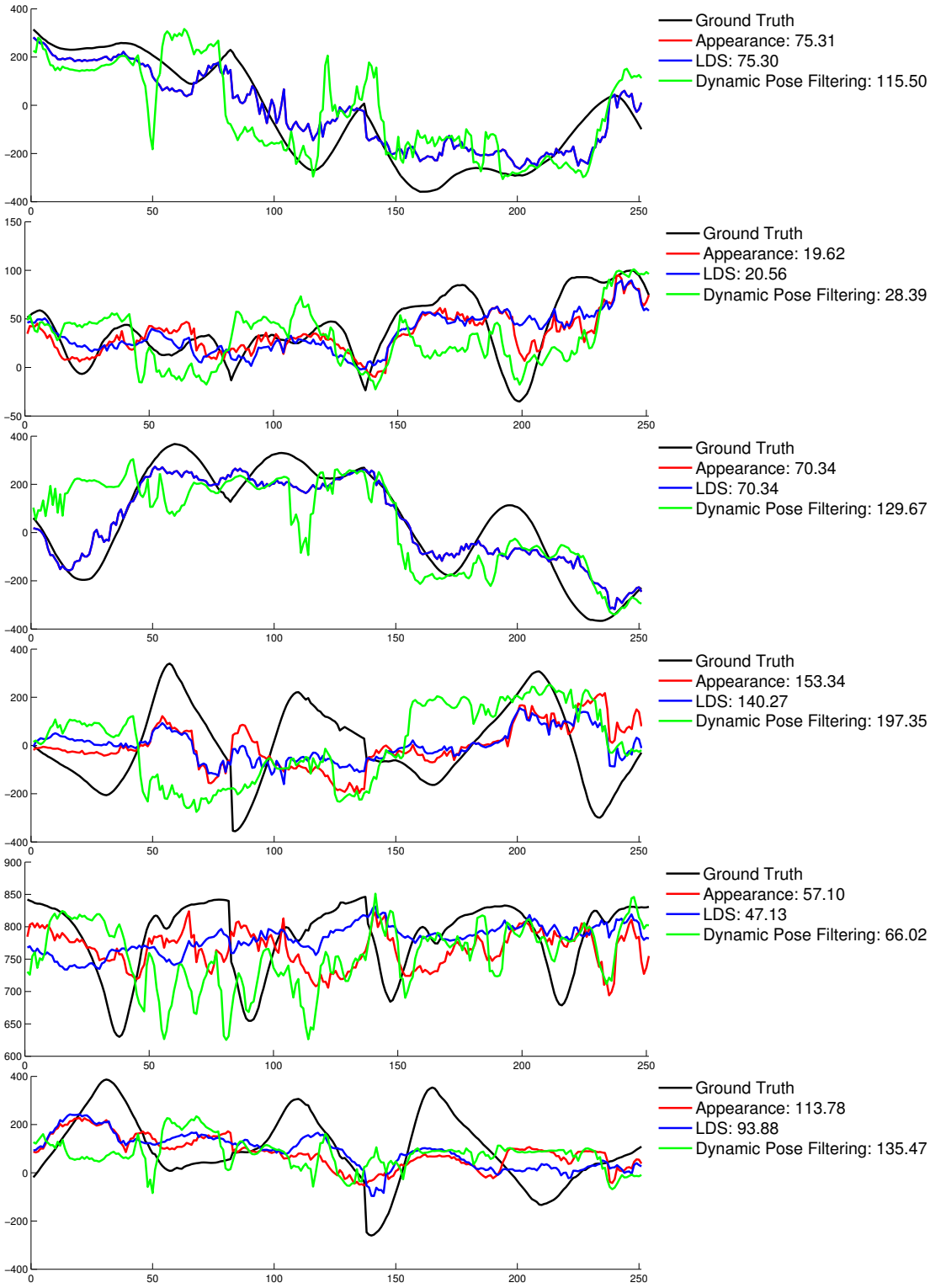

Figure 6.9: Joint position over time for the X, Y and Z axes of the left foot and left arm of the HumanEva Walking sequence. The black shows the ground truth, the red shows the appearance model, green is the dynamical pose filtering algorithm and blue is a linear dynamical system. Legend gives the mean absolute error for each model over this segment.

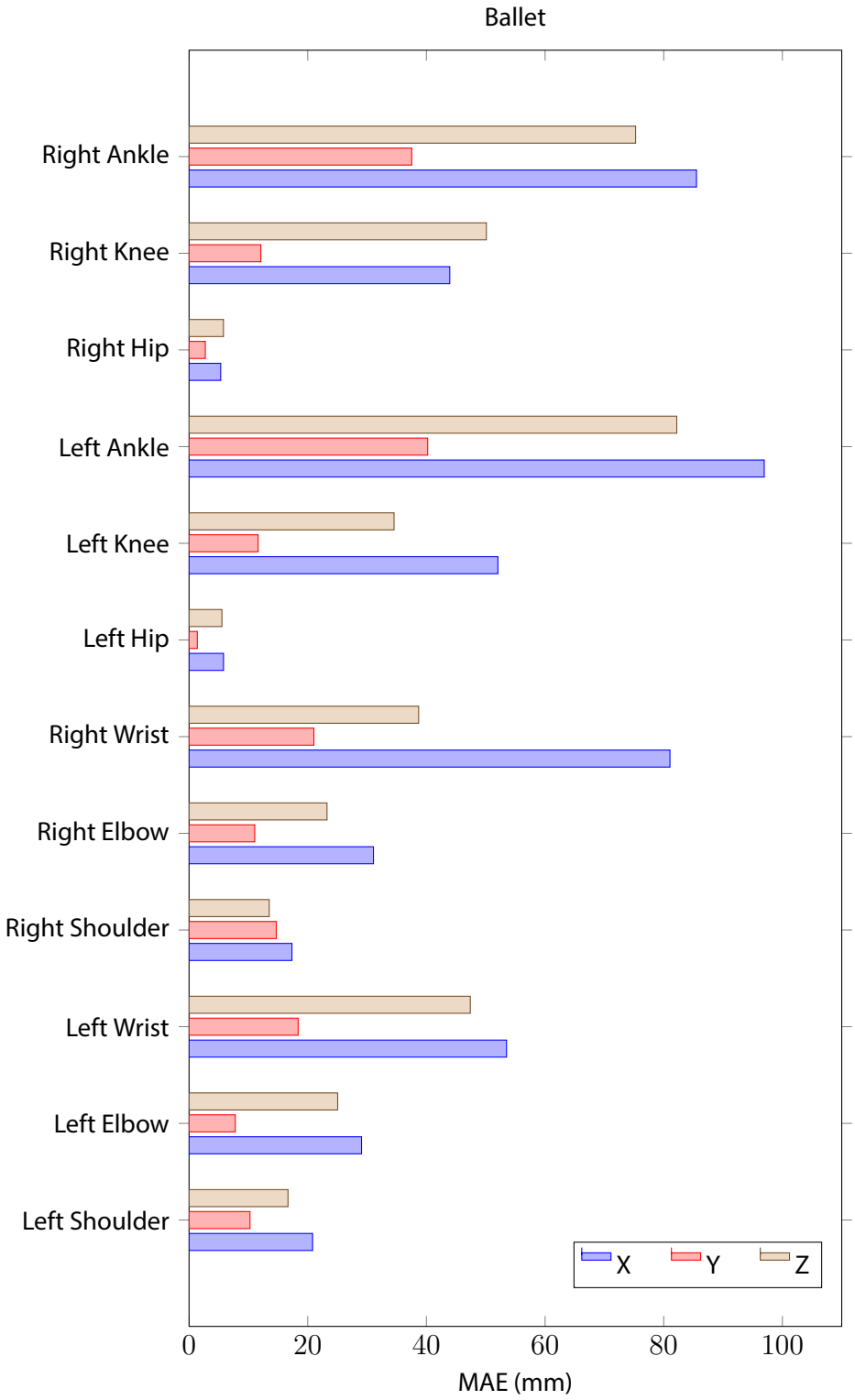

Figure 6.10: HumanEva walking, mean absolute error per joint. Large errors are seen on the subject's ankles whose movement can be highly non-linear.

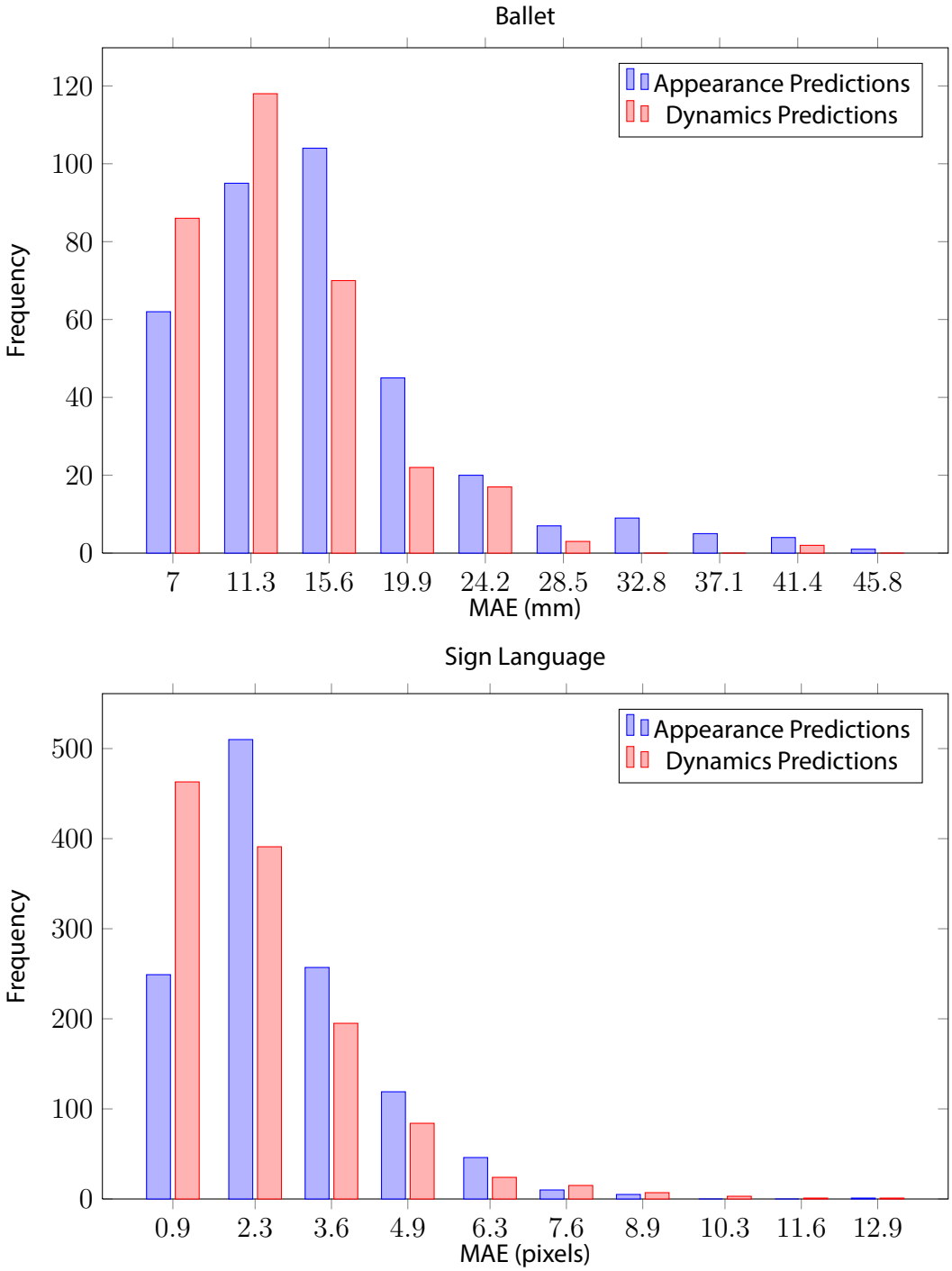

Figure 6.11: Jitter histograms for the Ballet and Sign Language data sets. We histogram the disparity between consecutive frames in a sequence to give a measure of how smooth a predicted pose sequence is. The x-axis gives the mean absolute error between consecutive predicted frames, and the y-axis gives the frequency of consecutive frames which fall into each error band. Errors above 50mm for ballet, and 17 pixels for sign language have been omitted to ensure good scale of the frequencies. We see that the disparities are lower on average for the dynamics predictions, showing greater temporal coherency.

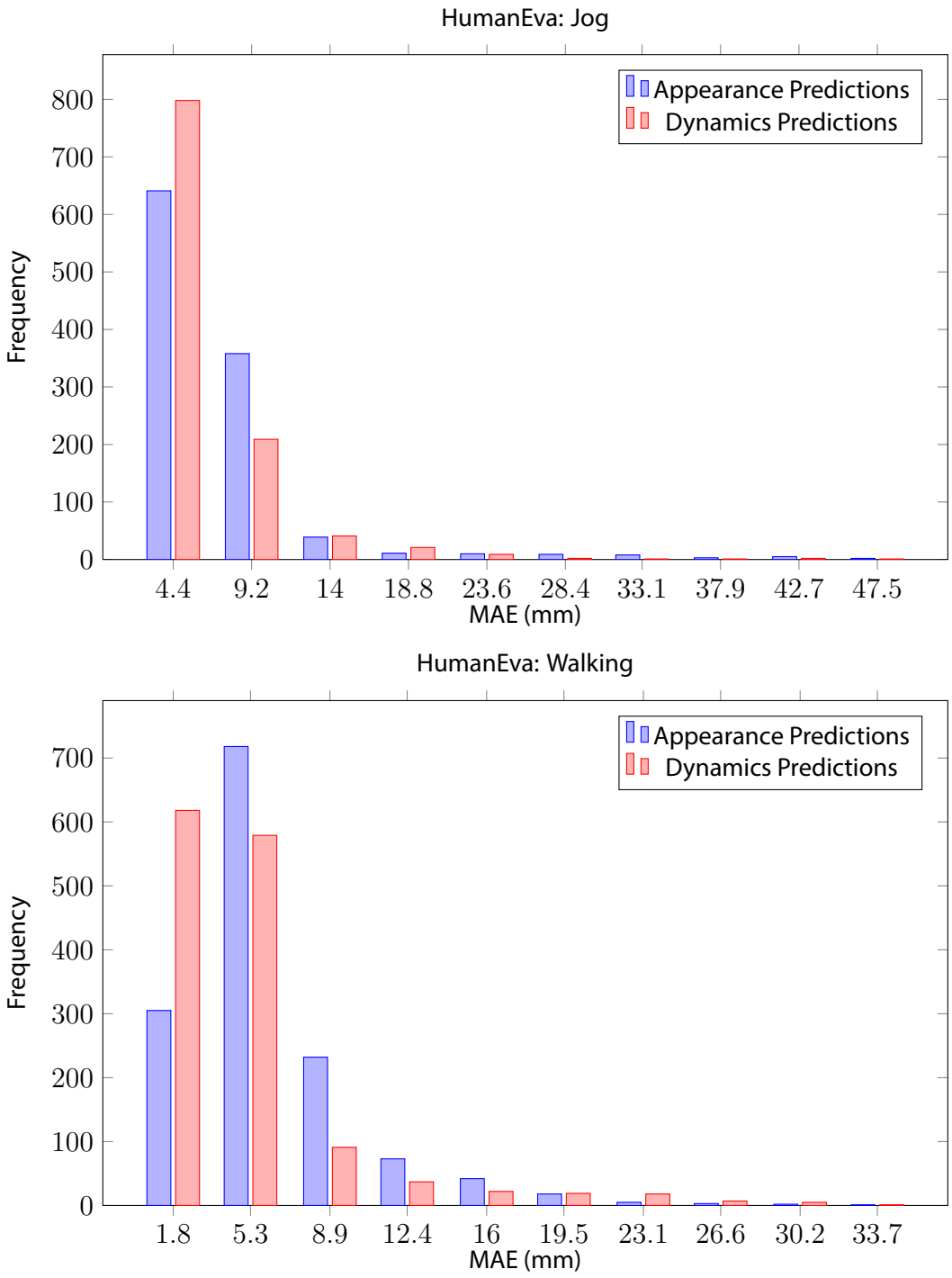

Figure 6.12: Jitter histograms for the HumanEva data set. We histogram the disparity between consecutive frames in a sequence to give a measure of how smooth a predicted pose sequence is. The x-axis gives the mean absolute error between consecutive predicted frames, and the y-axis gives the frequency of consecutive frames which fall into each error band. Errors above 50mm have been omitted to ensure good scale of the frequencies. We see that the disparities are lower on average for the dynamics predictions, showing greater temporal coherency.

7 Conclusion

In this thesis we have proposed new algorithms for using Gaussian processes for discriminative human pose estimation. Discriminative pose estimation attempts to infer the articulated pose of a subject directly from an extracted image feature. This allows fast inference to be achieved and flexible models which can be applied to a variety of data sets without modification. Models are built from large offline data sets which contain annotated pose images, allowing regression models to learn the mapping from image features to the pose space. In this thesis we have discussed how to represent articulated human pose and what image features are effective for capturing the image information relevant to human pose.

We show how to deploy Gaussian processes (GPs) in a mixture of experts framework in a way that allows Gaussian processes to scale to large data sets with high dimensional features. Large data sets are typical for discriminative pose estimation where varied training sets are required to build models which can handle a diverse range of poses. High dimensional features are required to represent visual clues which contain information relevant to a subject's pose. Gaussian processes are powerful regression techniques that can model non-linear functional mappings with accurate modelling of uncertainty. Their kernel formulation allows them to model large feature spaces by considering the training data as pairwise distances. They do not suffer from the problems of collapsing uncertainty as with other kernel linear models.

However, GPs have previously been limited to small data sets due to their $O(N^3)$ training complexity. Their uni-modal predictive distribution limits their ability to model the ambiguities and multi-modalities present in discriminative pose estimation. We show that by incorporating Gaussian processes in a mixture of experts model, they can scale to large data sets and model the multi-modal mapping between image and pose. The techniques outlined in this thesis are general and can be applied to any problem which requires non-linear and multi-modal regression.

Our first contribution is a mixture of Gaussian processes where we learn multiple Gaussian processes, each representing a local region of the pose space. This allows multi-modal data to be modelled by representing each mode with an individual GP. By limiting the size of each GP, we create a model which can scale linearly with the data set size, allowing ever larger problems to be tackled. We show how a logistic regression model can be used to give a weight to each expert's influence in the predictive distribution. We show that this algorithm is relatively insensitive to parameter choices, and gives state of the art performance compared to other discriminative models. This work has been published in [29].

Our second contribution extends this model to automatically optimise the size and location of each expert. Gaussian processes are able to accurately model the non-linear functions with consistent ambiguity. If these assumptions are violated through multi-modality or varying ambiguity, the Gaussian process will not model the distribution of the data accurately. Our proposed algorithm uses Gibbs sampling to learn a set of indicator variables which assign each training point to belong to an expert. The learnt indicators ensure each expert is trained with data which is well modelled by a single GP leading to a more accurate predictive distribution. We show that by learning the indicators in this way our model out performs our previous mixture of Gaussian processes model on human pose estimation data sets. This work has been published in [28].

Our final contribution is a dynamics framework for estimating a smooth pose sequence from a sequence of independent predictive distributions. Discriminative pose estimation makes predictions independently for each frame. Our algorithm uses a second order dynamics constraint to infer a smooth pose through a sequence of Gaussian mixture models – the predictive distribution of our mixture of Gaussian processes model. The algorithm considers the appearance observation of each expert in turn, combining it with a dynamics prediction formed by integrating out the poses from the previous frames. A dynamic programming algorithm is used to infer the optimal appearance expert for each frame in the sequence. We show that our algorithm is able to give a smooth estimate of human pose, correcting some mistakes made by our appearance model alone. We show that our model outperforms a baseline linear dynamical system. This work has been published in [28].

7.1 DISCUSSION

One of the primary limitations of discriminative human pose estimation is the availability of training data. Collecting training data is an expensive process. The ballet and sign language data sets used in this thesis have been manually annotated, a labour

intensive process which can often contain erroneous annotations. Data sets such as HumanEva have been captured using a motion capture system. This process enforces the subject to wear a black body suit for effective marker tracking – limiting the real-world applicability of the data. Recent work on the Microsoft Kinect platform has shown that synthetic data can be a useful tool for building real-world systems. However realistic synthetic data is far harder to generate for colour images of humans as opposed to the depth images used in their system.

The mixture of Gaussian process models we propose in this thesis are shown to be powerful regression techniques which give state of the art performance on human pose estimation data sets. By training each expert on a local region of pose space, we allow Gaussian processes to be applied to large scale problems, taking advantage of their accurate predictive distributions. As larger data sets become available it will be interesting to see how well the logistic regression gating model will scale. We have shown it to work well with 100 experts for data sets with up to 4400 training points, but it is unclear whether these models will scale up to data sets which require thousands of experts.

Our dynamics algorithm allows the accuracy of pose estimation to be improved by introducing a dynamics constraint. Although it does an effective job of smoothing the predicted pose giving better visual results, it does not have a significant effect on reducing the tracking error. One of the possible causes of these errors are incorrect predictions from the appearance model. If the correct pose for a frame isn't given a high likelihood by the predictive distribution then the dynamics model will make an incorrect prediction. The second order dynamics model gives a general representation of human motion, but does not model specific gestures or activities. This can be a desirable property the model can be applied to a wide range of activities without requiring training data for that specific activity or gesture. Any deviation from the second order linear prediction is modelled as uncertainty – relying on the appearance model to identify the correct pose. Modelling the subtle non-linearities in second order human pose would allow more accurate dynamics predictions to be made. However this is a challenging problem as such a model will have to support very fast predictions with large training data sets while giving an accurate predictive uncertainty.

7.2 FUTURE CHALLENGES

Improved gating network – as the available data sets grow, it is unclear how well a logistic regression model will scale to predicting hundreds of classes. As data sets grow, image features with larger dimensionality will be required to facilitate a greater variety of pose. This will cause the number of features relative to the expert sizes increase, resulting in

a more difficult learning problem for the logistic regression model.

Non-linear dynamics – although human motion is modelled reasonably well with second order linear dynamics, there are subtle non-linearities which lead to incorrect predictions. A regression model which is able to model these non-linearities while still retaining fast prediction and a stable predictive uncertainty would offer improved dynamics performance.

Bibliography

[1] A. Agarwal and B. Triggs. Hyperfeatures-multilevel local coding for visual recognition. *Lecture Notes in Computer Science*, 3951:30, 2006.

[2] A. Agarwal and B. Triggs. Recovering 3d human pose from monocular images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(1):44–58, Jan. 2006.

[3] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1014 –1021, june 2009.

[4] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios. Boostmap: An embedding method for efficient nearest neighbor retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(1):89–104, Jan. 2008.

[5] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning distance functions using equivalence relations. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, volume 20, page 11, 2003.

[6] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(4):509–522, Apr 2002.

[7] C. Bishop and M. Svensen. Bayesian hierarchical mixtures of experts. *Uncertainty in Artificial Intelligence*, 2003.

[8] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[9] C.M. Bishop and M.E. Tipping. Variational relevance vector machines. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 46–53. Citeseer, 2000.

[10] L. Bo and C. Sminchisescu. Twin Gaussian Processes for Structured Prediction. *International Journal of Computer Vision*, 2010.

[11] L. Bo, C. Sminchisescu, A. Kanaujia, and D. Metaxas. Fast algorithms for large scale conditional 3d prediction. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[12] Lubomir Bourdev and Jitendra Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1365 –1372, 29 2009-oct. 2 2009.

[13] M. Brand. Shadow puppetry. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1237 –1244 vol.2, 1999.

[14] M. Bray, E. Koller-Meier, NN Schraudolph, and L. Van Gool. Fast stochastic optimization for articulated structure tracking. *Image and Vision Computing*, 25(3):352–364, 2007.

[15] M. Bray, E. Koller-Meier, and L. Van Gool. Smart particle filtering for high-dimensional tracking. *Computer Vision and Image Understanding*, 106(1):116–129, 2007.

[16] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 8 –15, jun 1998.

[17] Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001. 10.1023/A:1010933404324.

[18] M.A. Brubaker, D.J. Fleet, and A. Hertzmann. Physics-based person tracking using the anthropomorphic walker. *International journal of computer vision*, 87(1):140–155, 2010.

[19] P. Buehler, MR Everingham, DP Huttenlocher, and A. Zisserman. Long term arm and hand tracking for continuous sign language TV broadcasts. In *Proceedings of the 19th British Machine Vision Conference*, pages 1105–1114. BMVA Press, 2008.

[20] J. Carreira and C. Sminchisescu. CPMC: Automatic Object Segmentation Using Constrained Parametric Min-Cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[21] Jixu Chen, Minyoung Kim, Yu Wang, and Qiang Ji. Switching gaussian process dynamic models for simultaneous composite motion tracking and recognition. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:2655–2662, 2009.

[22] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 1:886–893 vol. 1, June 2005.

[23] E. de Aguiar, C. Theobalt, C. Stoll, and H.-P. Seidel. Marker-less deformable mesh tracking for human shape and motion capture. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1 –8, june 2007.

[24] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 126 –133 vol.2, 2000.

[25] J. Drugowitsch and A.M. Barry. Generalised Mixtures of Experts, Independent Expert Training, and Learning Classifier Systems. Technical report, Technical Report 2007-12). University of Bath, UK, 2007.

[26] C. H. Ek, P. H. Torr, and N. D. Lawrence. Gaussian process latent variable models for human pose estimation. *A. Popescu-Belis, S. Renals and H. Bourlard (eds) Machine Learning for Multimodal Interaction (MLMI 2007), Springer-Verlag, Brno, Czech Republic*, pages 132–143, 2007.

[27] P.F. Felzenszwalb and D.P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005.

[28] M. Fergie and A. Galata. Dynamical pose filtering for mixtures of gaussian processes. In *Proceedings of the British Machine Vision Conference*, 2012.

[29] Martin Fergie and Aphrodite Galata. Local gaussian processes for pose recognition from noisy inputs. In *Proceedings of the British Machine Vision Conference*, pages 98.1–98.11. BMVA Press, 2010. doi:10.5244/C.24.98.

[30] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[31] Y. Freund and R. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer, 1995.

[32] J. Gall, C. Stoll, E. de Aguiar, C. Theobalt, B. Rosenhahn, and H.-P. Seidel. Motion capture using joint skeleton tracking and surface estimation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1746 –1753, june 2009.

[33] D.M. Gavrila and L.S. Davis. 3-d model-based tracking of humans in action: a multi-view approach. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on*, pages 73 –80, jun 1996.

[34] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon. Efficient regression of general-activity human poses from depth images. *International Conference on Computer Vision (ICCV)*, 2011.

[35] Shaobo Hou, A. Galata, F. Caillette, N. Thacker, and P. Bromiley. Real-time body tracking using a gaussian process latent variable model. *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct. 2007.

[36] N. Howe, M. Leventon, and W. Freeman. Bayesian reconstruction of 3d human motion from single-camera video. In *Neural Information Processing Systems*, volume 1999, page 1. Cambridge, MA, 1999.

[37] C. Ionescu, Fuxin Li, and C. Sminchisescu. Latent structured models for human pose estimation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2220 –2227, nov. 2011.

[38] Michael Isard and Andrew Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, August 1998.

[39] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Proceedings of the British Machine Vision Conference*, pages 12.1–12.11. BMVA Press, 2010. doi:10.5244/C.24.12.

[40] M.I. Jordan and R.A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural computation*, 6(2):181–214, 1994.

[41] L. Kakadiaris and D. Metaxas. Model-based estimation of 3d human motion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(12):1453 –1459, dec 2000.

[42] A. Kanaujia, C. Sminchisescu, and D. Metaxas. Semi-supervised hierarchical models for 3d human pose reconstruction. *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007.

[43] Atul Kanaujia and Dimitris Metaxas. Learning ambiguities using bayesian mixture of experts. *Tools with Artificial Intelligence, 2006. ICTAI '06. 18th IEEE International Conference on*, pages 436–440, Nov. 2006.

[44] N.D. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In *Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference*, page 329. The MIT Press, 2004.

[45] M. Levoy. The digital michelangelo project. In *3-D Digital Imaging and Modeling, 1999. Proceedings. Second International Conference on*, pages 2 –11, 1999.

[46] Y. Liu, C. Stoll, J. Gall, H.P. Seidel, and C. Theobalt. Markerless motion capture of interacting characters using multi-view image segmentation. *Computer Vision and Pattern Recognition (CVPR)*, 2011.

[47] D.G. Lowe. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157 vol.2, 1999.

[48] E. Meeds and S. Osindero. An alternative infinite mixture of gaussian process experts. *Advances in Neural Information Processing Systems*, 18:883, 2006.

[49] Roland Memisevic, Leonid Sigal, and David J. Fleet. Shared kernel information embedding for discriminative inference. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(4):778 –790, april 2012.

[50] I. Mikic, M. Trivedi, E. Hunter, and P. Cosman. Articulated body posture estimation from multi-camera voxel data. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–455 – I–460 vol.1, 2001.

[51] D.D. Morris and J.M. Rehg. Singularity analysis for articulated object tracking. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 289–296. IEEE, 1998.

[52] J. Müller and M. Arens. Human pose estimation with implicit shape models. In *Proceedings of the first ACM international workshop on Analysis and retrieval of tracked events and motion in imagery streams*, pages 9–14. ACM, 2010.

[53] Huazhong Ning, Wei Xu, Yihong Gong, and T. Huang. Discriminative learning of visual words for 3d human pose estimation. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[54] T Pfister, J Charles, M Everingham, and A Zisserman. Automatic and efficient long term arm and hand tracking for continuous sign language tv broadcasts. *BMVC*, 2012.

[55] R. Plankers and P. Fua. Articulated soft objects for video-based body modeling. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 394 –401 vol.1, 2001.

[56] R. Plankers and P. Fua. Articulated soft objects for multiview shape and motion capture. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(9):1182 – 1187, sept. 2003.

[57] D. Ramanan. Learning to parse images of articulated bodies. *Advances in Neural Information Processing Systems*, 19:1129, 2007.

[58] L. Raskin, E. Rivlin, and M. Rudzsky. Using gaussian process annealing particle filter for 3d human tracking. *EURASIP Journal on Advances in Signal Processing*, 2008, 2008.

[59] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[60] C.E. Rasmussen. The infinite Gaussian mixture model. *Advances in neural information processing systems*, 12:554–560, 2000.

[61] C.E. Rasmussen and Z. Ghahramani. Infinite mixtures of Gaussian process experts. In *Advances in neural information processing systems 14: proceedings of the 2001 conference*, page 881. MIT Press, 2002.

[62] J.M. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 612 –617, jun 1995.

[63] S. Romdhani, P. Torr, B. Schölkopf, and A. Blake. Efficient face detection by a cascaded support–vector machine expansion. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 460(2051):3283–3297, 2004.

[64] B. Rosenhahn, U. Kersting, K. Powell, R. Klette, G. Klette, and H.P. Seidel. A system for articulated tracking incorporating a clothing model. *Machine Vision and Applications*, 18(1):25–40, 2007.

[65] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 309–314. ACM, 2004.

[66] S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding, 2000.

[67] J. Saboune and F. Charpillet. Using interval particle filtering for marker less 3d human motion capture. In *Tools with Artificial Intelligence, 2005. ICTAI 05. 17th IEEE International Conference on*, pages 7 pp. –627, nov. 2005.

[68] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 519 – 528, june 2006.

[69] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2:994–1000 vol. 2, June 2005.

[70] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge Univ Pr, 2004.

[71] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'11), Colorado Springs, USA*, 2011.

[72] Hedvig Sidenbladh, Michael Black, and David Fleet. Stochastic tracking of 3d human figures using 2d image motion. In David Vernon, editor, *Computer Vision — ECCV 2000*, volume 1843 of *Lecture Notes in Computer Science*, pages 702–718. Springer Berlin / Heidelberg, 2000.

[73] L. Sigal, A. Balan, and M.J. Black. Combined discriminative and generative articulated pose and non-rigid shape estimation. *Advances in Neural Information Processing Systems*, 2007.

[74] L. Sigal and M.J. Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. *Brown Univertsity TR*, 2006.

[75] C. Sminchisescu and A. Jepson. Variational mixture smoothing for non-linear dynamical systems. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–608 – II–615 Vol.2, june-2 july 2004.

[76] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Conditional models for contextual human motion recognition. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1808–1815. IEEE, 2005.

[77] C. Sminchisescu, A. Kanaujia, Zhiguo Li, and D. Metaxas. Discriminative density propagation for 3d human motion estimation. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 1:390–397 vol. 1, June 2005.

[78] C. Sminchisescu, A. Kanaujia, and D. Metaxas@inproceedingspitsikalis2010data, title=Data-Driven Sub-Units and Modeling Structure for Continuous Sign Language Recognition with Multiple Cues, author=Pitsikalis, V. and Theodorakis, S. and Maragos, P., booktitle=LREC Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies, volume=1, number=2, year=2010 . Learning joint top-down and bottom-up processes for 3d visual inference. *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2:1743–1752, 2006.

[79] C. Sminchisescu and B. Triggs. Covariance scaled sampling for monocular 3d body tracking. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–447. IEEE, 2001.

[80] B. Stenger, P.R.S. Mendonca, and R. 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 – II–315 vol.2, 2001.

[81] B. Stenger, A. Thayananthan, P.H.S. Torr, and R. Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(9):1372–1384, Sept. 2006.

[82] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[83] A. Thayananthan, R. Navaratnam, B. Stenger, P. Torr, and R. Cipolla. Multivariate relevance vector machines for tracking. *Computer Vision–ECCV 2006*, pages 124–138, 2006.

[84] M.E. Tipping and C.M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, pages 611–622, 1999.

[85] R. Urtasun and T. Darrell. Sparse probabilistic regression for activity-independent human pose inference. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[86] R. Urtasun, D.J. Fleet, and P. Fua. Monocular 3d tracking of the golf swing. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 932 – 938 vol. 2, june 2005.

[87] R. Urtasun, D.J. Fleet, and P. Fua. 3d people tracking with gaussian process dynamical models. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 238–245, June 2006.

[88] Raquel Urtasun, David J. Fleet, Aaron Hertzmann, and Pascal Fua. Priors for people tracking from small training sets. *Computer Vision, IEEE International Conference on*, 1:403–410, 2005.

[89] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. In *ACM Transactions on Graphics (TOG)*, volume 27, page 97. ACM, 2008.

[90] S. Wachter and H.-H. Nagel. Tracking of persons in monocular image sequences. In *Nonrigid and Articulated Motion Workshop, 1997. Proceedings., IEEE*, pages 2 –9, jun 1997.

[91] J.M. Wang, D.J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):283–298, feb. 2008.

[92] Yang Wang, Duan Tran, and Zicheng Liao. Learning hierarchical poselets for human parsing. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1705 –1712, june 2011.

[93] Xu Zhao, Huazhong Ning, Yuncai Liu, and T. Huang. Discriminative estimation of 3d human pose using gaussian processes. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4, Dec. 2008.

[94] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 28–31. IEEE, 2004.