From Images and Sounds to Face Localization and Tracking

Vincent Drouard

Communauté UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Mathématiques et Informatique**

Arrêté ministériel : 25 Mai 2016

Présentée par

Vincent Drouard

Thèse dirigée par **Radu Horaud**

préparée au sein de **l'Inria Grenoble Rhône-Alpes**
et de **l'École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique**

Localisation et suivi de visages à partir d'images et de sons

Une approche Bayésienne temporelle et commutative

Thèse soutenue publiquement le **18 Décembre 2017**,
devant le jury composé de :

Pr. Florence Forbes
Inria Grenoble Rhône-Alpes, Présidente
Pr. Manuel J. Marin-Jimenez
Universidad de Córdoba, Rapporteur
Pr. Olivier Alata
Université Jean Monnet, Saint-Etienne, Rapporteur
Pr. Vincent Lepetit
Université de Bordeaux, Examinateur
Dr. Xavier Alameda-Pineda
Inria Grenoble Rhône-Alpes, Examinateur
Pr. Radu Horaud
Inria Grenoble Rhône-Alpes, Directeur de thèse

From Images and Sounds to Face Localization and Tracking:
A Switching Dynamical Bayesian Framework

Vincent Drouard

December 19, 2017

Abstract

In this thesis, we address the well-known problem of *head-pose estimation* in the context of *human-robot interaction* (HRI). We accomplish this task in a two step approach. First, we focus on the estimation of the head pose from visual features. We design features that could represent the face under different orientations and various resolutions in the image. The resulting is a high-dimensional representation of a face from an RGB image. Inspired from [Deleforge 15] we propose to solve the head-pose estimation problem by building a link between the head-pose parameters and the high-dimensional features perceived by a camera. This link is learned using a high-to-low probabilistic regression built using probabilistic a mixture of affine transformations. With respect to classic head-pose estimation methods we extend the head-pose parameters by adding some variables to take into account variety in the observations (e.g. misaligned face bounding-box), to obtain a robust method under realistic conditions. Evaluation of the methods shows that our approach achieve better results than classic regression methods and similar results than state of the art methods in head pose that use additional cues to estimate the head pose (e.g depth information). Secondly, we propose a temporal model by using tracker ability to combine information from both the present and the past. Our aim through this is to give a smoother estimation output, and to correct oscillations between two consecutives independent observations. The proposed approach embeds the previous regression into a temporal filtering framework. This extension is part of the family of switching dynamic models and keeps all the advantages of the mixture of affine regressions used. Overall the proposed tracker gives a more accurate and smoother estimation of the head pose on a video sequence. In addition, the proposed switching dynamic model gives better results than standard tracking models such as Kalman filter. While being applied to the head-pose estimation problem the methodology presented in this thesis is really general and can be used to solve various regression and tracking problems, e.g. we applied it to the tracking of a sound source in an image.

Résumé

Dans cette thèse, nous abordons le problème de l'estimation de pose de visage dans le contexte des interactions homme-robot. Nous abordons la résolution de cette tâche à l'aide d'une approche en deux étapes. Tout d'abord en nous inspirant de [Deleforge 15], nous proposons une nouvelle façon d'estimer la pose d'un visage, en apprenant un lien entre deux espaces, l'espace des paramètres de pose et un espace de grande dimension représentant les observations perçues par une caméra. L'apprentissage de ce lien se fait à l'aide d'une approche probabiliste, utilisant un mélange de regressions affines. Par rapport aux méthodes d'estimation de pose de visage déjà existantes, nous incorporons de nouvelles informations à l'espace des paramètres de pose, ces additions sont nécessaires afin de pouvoir prendre en compte la diversité des observations, comme les differents visages et expressions mais aussi les décalages entre les positions des visages détectés et leurs positions réelles, cela permet d'avoir une méthode robuste aux conditions réelles. Les évaluations ont montrées que cette méthode permettait d'avoir de meilleurs résultats que les méthodes de regression standard et des résultats similaires aux méthodes de l'état de l'art qui pour certaines utilisent plus d'informations, comme la profondeur, pour estimer la pose. Dans un second temps, nous développons un modèle temporel qui utilise les capacités des traqueurs pour combiner l'information du présent avec celle du passé. Le but à travers cela est de produire une estimation de la pose plus lisse dans le temps, mais aussi de corriger les oscillations entre deux estimations consécutives indépendantes. Le modèle proposé intègre le précédent modèle de régression dans une structure de filtrage de Kalman. Cette extension fait partie de la famille des modèles dynamiques commutatifs et garde tous les avantages du mélange de regressions affines précédent. Globalement, le modèle temporel proposé permet d'obtenir des estimations de pose plus précises et plus lisses sur une vidéo. Le modèle dynamique commutatif donne de meilleurs résultats qu'un modèle de suivi utilsant un filtre de Kalman standard. Bien qu'appliqué à l'estimation de pose de visage le modèle presenté dans cette thèse est très général et peut être utilisé pour résoudre d'autres problèmes de régressions et de suivis.

Περίληψη

Στην παρούσα διδακτορική διατριβή, θίγουμε το πρόβλημα του υπολογισμού της τοποθέτησης του προσώπου στα πλαίσια των αλληλεπιδράσεων ανθρώπου-ρομπότ. Θίγουμε τη λύση αυτής της εργασίας μέσω μίας προσέγγισης σε δύο στάδια. Εμπνεόμενοι αρχικά, από τον [Deleforge 15], προτείνουμε έναν καινούριο τρόπο υπολογισμού της τοποθέτησης ενός προσώπου διδάσκοντας μία σχέση ανάμεσα σε δύο χώρους, τον χώρο των παραμέτρων της τοποθέτησης και ένα χώρο μεγάλης διάστασης που αντιπροσωπεύει τις παρατηρήσεις που γίνονται αντιληπτές μέσω μίας κάμερας. Η μαθητεία αυτής της σχέσης γίνεται με τη βοήθεια μίας πιθανολογικής προσέγγισης, χρησιμοποιώντας ένα μείγμα παλινδρομήσεων συγγένειας. Σχετικά με τις υπάρχουσες μεθόδους υπολογισμού της τοποθέτησης του προσώπου, ενσωματώνουμε καινούριες πληροφορίες στο χώρο των παραμέτρων της τοποθέτησης, αυτές οι προσθήκες είναι αναγκαίες ώστε να μπορέσουμε να λάβουμε υπόψη την ποικιλία των παρατηρήσεων, όπως τα διαφορετικά πρόσωπα και οι εκφράσεις αλλά επίσης και τις διαφορές ανάμεσα στις θέσεις των προσώπων που έχουν ανιχνευτεί και τις πραγματικές θέσεις αυτών, αυτό επιτρέπει να έχουμε μία εύρωστη μέθοδο σε πραγματικές συνθήκες. Οι αξιολογήσεις έχουν δείξει ότι αυτή η μέθοδος επέτρεπε να έχουμε καλύτερα αποτελέσματα από άλλες μεθόδους κανονικής μείωσης και παρόμοια αποτελέσματα από την εφαρμογή μεθόδων μελέτης της τρέχουσας κατάστασης που για ορισμένους, χρησιμοποιούν περισσότερες πληροφορίες, όπως το βάθος, για να υπολογίσουν την τοποθέτηση. Κατά δεύτερον, αναπτύσσουμε ένα χρονικό μοντέλο που χρησιμοποιεί τις ικανότητες ανιχνευτών για να συνδυάσει την παρούσα πληροφορία με την παρελθούσα. Ο σκοπός ο οποίος επιδιώκεται μέσω αυτού του μοντέλου είναι να παραχθεί μία εκτίμηση της τοποθέτησης περισσότερο λεία μέσα στο χρόνο, αλλά και να διορθωθούν επίσης οι ταλαντώσεις μεταξύ δύο ανεξάρτητων διαδοχικών εκτιμήσεων.Το μοντέλο που προτείνεται, ενσωματώνει το προηγούμενο μοντέλο ύφεσης μέσα σε μία δομή φιλτραρίσματος του Kalman. Αυτή η επέκταση ανήκει στην οικογένεια των δυναμικών αντιμεταθετικών μοντέλων και διατηρεί όλα τα πλεονεκτήματα του προηγούμενου μείγματος εξευγενισμένων υφέσεων. Γενικά, το προτεινόμενο χρονικό μοντέλο επιτρέπει την απόκτηση εκτιμήσεων τοποθέτησης με περισότερη ακρίβεια και πιο λείες σε ένα βίντεο. Το δυναμικό αντιμεταθετικό μοντέλο παρέχει καλύτερα αποτελέσματα από ένα μοντέλο παρακολούθησης που χρησιμοποιεί ένα κανονικό φίλτρο του Kalman. Αν και έχει εφαρμοστεί στην εκτίμηση της τοποθέτησης του προσώπου το μοντέλο που παρουσιάζεται στην παρούσα διδακτορική διατριβή είναι πολύ γενικό και μπορεί να χρησιμοποιηθεί για την επίλυση κι άλλων προβλημάτων παλινδρομήσης και παρακολουθήσεων.

ACKNOWLEDGMENT

Tout d'abord je dois un grand merci a Radu, pour m'avoir guider pendant ces trois années. Pour avoir transformé mes brouillons en article de qualités. Ton soutien et ta passion dans ce que tu fais ont été une grande source de motivation et je suis fier d'avoir apris et travaillé à tes côtés.

Silèye, Γεώργιο, Antoine et Xavi pour avoir eu la chance de travailler à vos côtés.

Je tiens aussi à remercier les membres de l'équipe Perception que j'ai cottoyé durant ces trois années, Quentin, Fabien, Stéphane, Benoît, Pablo, Laurent, 李晓飞, Bastien, Guillaume, Guyllôme, Sylvain et ceux que j'ai pu oubliés.

Nathalie pour t'être toujours occupée de tout et de nous avoir permis d'avoir des vacances aux quatres coins du monde.

Soraya pour m'avoir ~~cassé les~~ poussé pour écrire mon manuscrit, tes conseils pour la rédaction ont été précieux.

Χαρά pour m'avoir aidé à traduire mon résumé, ευχαριστώ.

Διόνυσσο, Israel, 张弘, ΄Αντρια , 班雨桐 and Daan for all the fun.

΄Αντρεα because you were here to support me everyday and for all the many good things that happened during these 3 years, thank you.

A great thank you to my examining committee for thoroughly evaluating this manuscript.

Enfin pour finir, merci à mes parents, à Anne et à Marie pour leur soutien depuis le début.

Je vis la vie que j'aurai pas

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In a social gathering, the human has the ability to extract a lot of information from visual auditive sense, e.g. where the people are, who are they, who are the speakers, who are they talking to, and more in order to determine the complete status of everyone. While this for a human is natural, this task for a robot is more challenging. How can these information be extracted from audio and video signals. A lot of works were done in order to find different representation of these signals in order to make the extraction of these information easier, e.g. using frequency representation of an audio signal, using carateristic point in image or its gradient. The second task is to a build model that learn how to extract the relevant information from these features. A lot of approaches exist, regression technics, probabilistic inference and more recently deep learning methods. In this thesis we tackle the challenge of estimating and tracking the head pose from videos. To do so we propose a methodology for tracking the head pose from high-dimensional observations combining regression, probabilistic inference and tracking.

This chapter presents the context of this thesis, the inspiration to this work, the contribution of this thesis and, at the end, the organization of the rest of the manuscript.

1.1 THE GLOBAL PROJECT

This thesis was part of the Vision and Hearing In Action (VHIA) project overseen by Radu Horaud, it's a joint work among the members of the Perception team of the INRIA Grenoble. The VHIA project has for objective to give humanoid companion robots perception and knowledge about their social environment, i.e. to give a robot all the information about the people around it in order to be able to understand and communicate. Understanding of the social environment for a social robot is the root for being able to interact with its surrounding, and has not to be neglected and to be carefully handle. This is achieved by building models that are able to extract from audio and visual cues the specific required *low-level* information, as the number of people in the scene, their speech face orientation in order to combine them, and thus to obtain *high-level* information as the main speaker the object/person of interest. This in the end gives a robot the ability to provide the proper answer and to express the correct emotion.

1.2 INSPIRATION

The inspiration of the work done in this thesis comes from the work of Antoine Deleforge, a former PhD student of the Perception team. During his thesis he proposed a novel method to map two spaces with different dimensions using mixture of regression. The method was designed to map a high dimensional space of observations onto a low-dimensional space representing the object state space. He applied it to sound source localization in images from binaural features. Using features from a pair of microphones, that are high-dimensional, he managed to build a mapping between these features and their associated sound source position in the image plane. Starting from this idea, we asked ourself with my supervisor Radu Horaud if a mapping could be done in a similar way, but in a different context to solve other problems since its formulation is generic and is independent to the type of data used as input and output. We were interested in building a model for head-pose estimation, in the context of human-robot interaction, using RGB images from a robot camera. We wished also to extend this work to make it more robust, we thought that adding temporality would be a great extension and contribution to it.

1.3 PROBLEM OVERVIEW

The problem we wish to solve is to infer the face orientation of people, from camera images. This is challenging in many ways. First from the context of the work, the model has to be applied onto the Humanoid robot Nao, manufactured by Aldebaran robotics. The Nao robot has a single camera and does not provide high resolution images, thus the resolution of the faces in the images are not of high quality. In this context methods using landmarks detection to estimate the head pose are not good to use. The second challenging part of head-pose estimation methods is that they need to be able to provide accurate estimation in various cases, different background, different lighting, faces (glasses, expressions pilosity and all the possible variations that human offers us) and also various image resolutions. Finally the robot has to be able to infer these informations in real time, thus the full method from the detection must be fast enough to work online. We need to build a model that can infer accurately and quickly the head pose, and also that does not require a lot of resources to work.

1.4 CONTRIBUTIONS OF THIS THESIS

The objective of this thesis was to build probabilistic models in the context of human-robot social interactions. Toward this goal we focused first on the estimation and the tracking of the head pose and in a second time on the tracking of a speaker because the model is easily applicable to various applications thanks to its generic formulation. Why focusing on these two problems ? In social interaction these two information are important because they can lead to an understanding of the status of social gathering. Knowing who/where is the speaker through sound source tracking and where are the

people looking at through their head-pose orientation. Head pose is a crucial information to be able to estimate the visual focus of attention.

PROBABILISTIC MAPPING FOR HEAD-POSE ESTIMATION

In Chapter 2, starting from the work of [Deleforge 14], that uses probabilistic regression for the localization of sound source in an image, we build a model for head-pose estimation using the same principle. We explore the possibility offered by the model to make the estimation robust to different realistic conditions that could appear in a real interaction between a human and a robot. The main contribution was to be able to jointly estimate the head-pose parameters and also the offset of the localization when the detection is not properly aligned onto the face. The outcome of this idea is an iterative algorithm that refined the position while estimating the head-pose parameters. Through this we demonstrate the existence of a mapping between the image space and the pose parameters and face position space. We achieved similar or better results than state-of-the-art methods on three datasets for head-pose estimation. This work was published in [Drouard 15] and [Drouard 17b].

PROBABILISTIC TRACKING FOR HEAD-POSE AND SOUND-SOURCE ESTIMATION

In Chapter 3, we present an extension of the model used for head-estimation, for temporal tracking. During the experimentations on the regression model we realized that the output estimation of the head pose of a person in a temporal sequence was not smooth and that sometimes the estimation can be far from the ground truth, this is due to some error of localization and some perturbations in the image that could alter the features. We proposed a temporal model that combined the previous probabilistic regression with a temporal model. The resulting can be seen as a mixture of Kalman filters with the possibility to switch from one Kalman filter to another one. The switching makes the model intractable with time because the number of components in the mixture model increases too much at each time step. To overcome this we proposed two approaches to contain the number of components. The first one is a component merging approach. The second one is a variational approximation that fixed the number of components for each time, thus avoiding it to increase. We compare these two methods to express their advantages and drawbacks and compare them to other tracking methods. This work was published in [Drouard 17a].

TEMPORAL MODEL FOR SPEAKER TRACKING

In Chapter 4, we present a tracking model for sound source direction of arrival tracking. The model is based on the tracking model presented in Chapter 3 and the sound source direction of arrival estimation model of [Deleforge 14]. Natural speech signal time-frequency representation being sparse (i.e. a lot of inactive time-frequency point with no energy), we adjusted the observation model of the tracking model using the observation model of [Deleforge 14]. The final model is able to track object state (in this case the direction of arrival of a sound) from a series of sparse observations.

SOME WORK ON APPLICATION OF SPEAKER-LOCALIZATION USING HUMANOID ROBOT NAO

In Chapter 5, we present our algorithm for speaker localization with the Nao robot. This work was made with the engineers of the Perception team. The idea of this was to be able to use Matlab to run algorithm with Nao. We developed the algorithm to estimate the speaker localization, using the cameras and the microphones of the Nao, by combining sound source localization with face detection. This demo was part of the EARS (Embodied Audition for Robots) project, joint european project where the team Perception was involved with research laboratories from United-Kingdom, Germany, Israel and also Aldebaran robotics (the manufacturer of the NAO robot). Out of this the engineer of the Perception team developed the NaoLab software. This software allows people to easily develop and run algorithms with the Nao robot using various programming languages, e.g. Python, C++ or Matlab. This demo was part of the publication [Badeig 15].

1.5 ORGANIZATION OF THIS MANUSCRIPT

The rest of the manuscript is organized in three core chapters, each one describing a contribution of the thesis as described in previous section. Some conclusions are drawn in Chapter 6. Appendix explaining some derivations of the models, the publications related to this thesis and the list of references are in backmatter. Enjoy your reading.

CHAPTER 2

HEAD-POSE ESTIMATION

The model for head-posed estimation developed during this thesis is inspired from the probabilistic method for sound localization by [Deleforge 14]. The model is based on a mixture of local regression embedded in a probabilistic framework, training of the model parameters is done using pairs of observations and associated head-poses. The model offers the possibility to have partially latent output during the training. Indeed, the output variable of the model is assumed to be a concatenation of the head-pose parameters and some other values that can catch some variations in the images that could affect the estimation of the head pose. The idea was to build a robust model that could compensate some issues from images such as light conditions and also face bounding box detection. This model goes a bit further than just a classic head-pose estimation as the first experiments we performed revealed that the face detection had an influence on the estimation of the head-pose. Indeed, the detection is not always properly aligned onto a face. To correct this, we extended the output variable to have a detection refinement part to correct the misalignment. The core of this chapter is organized in five sections. First we define what is head-pose estimation in the context of computer vision and this thesis, we cover a part of the literature about this. Then we evaluate the efficiency of the methods on several datasets and compare to several *state-of-the-art* methods. Finally we draw some conclusions.

2.1 WHAT IS THE HEAD-POSE ESTIMATION

In computer vision, head-pose defines the orientation of a person's head with respect to a coordinate system. Here the coordinate system is defined by a camera, which leads to a pretty wide range of possible orientations. In general when we talk about head pose we refer to 2D or 3D head-pose, the dimension of the vector that contains the angles that define the orientation of a face. These parameters correspond to three angles, namely pitch (top to bottom rotation of the face), yaw (left to right) and roll (autorotation of the face), see Fig 2.1.

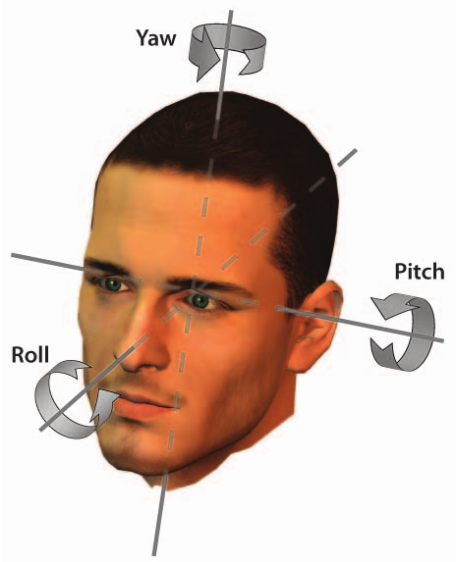

Figure 2.1: Representation of the head pose with the three possible egocentric rotations, source: [Murphy-Chutorian 09]

2.2 FROM IMAGE TO HEAD POSE: SEVERAL APPROACHES (RELATED WORK)

A complete review about head-pose estimation was published by [Murphy-Chutorian 09] few years ago. We suggest any reader to look at this paper for a complete coverage of the head pose literature until 2009. Here, we only focus on what has been done since. New technics were developed and new data became available since then like 3D data based methods and methods based on neural networks. Head-pose methods can be grouped into four categories: methods based on depth images, methods based on manifold learning, methods based on regression and those based on neural networks and especially using convolutive neural networks (CNN).

§ Depth images

The recent advent of depth cameras enabled fast development of depth-based head-pose methods. Depth data are more reliable as they allow to overcome some of the drawbacks of RGB data, such as illumination problems and facial landmarks detection, which is more reliable. One of the first methods using depth data was introduced by [Seemann 04]. In this method, a depth map of the head region is combined with a color histogram and used to train a neural network. [Fanelli 13] used random forest regression to estimate both the head pose and facial landmarks. [Peng 15] also used random forests in their study in which RGB scale-invariant feature transform (SIFT) descriptors are combined with 3D histogram of oriented gradients (HOG) descriptors. Finally, it should be highlighted that depth information is merely used to disambiguate photometric data and that depth data

used alone for head-pose estimation is not as effective as combining with RGB information.

§ Manifold learning

Several authors suggested to use manifold learning for head-pose estimation. Manifold learning consists in finding a low-dimensional output space of head poses from a high-dimensional input space of feature vectors. The inconvenient of Manifold learning is that the output variables do not necessarily correspond to the pose angles, which implies that learning, in a supervised way the mapping between the manifold-learning output and the desired space spanned by the pose parameters is also necessary. This has been achieved in various ways, e.g. [Srinivasan 02, Raytchev 04, Hu 05, Li 07, BenAbdelkader 10, Foytik 13, Sundararajan 15]. To conclude, these two step methods suffer from the fact that unsupervised manifold-learning techniques do not guarantee that the predicted output space contains the information needed for head pose.

§ Regression

Among the regression methods used for head pose are Gaussian process regression (GPR) [Marin-Jimenez 14], support vector regression (SVR) [Murphy-Chutorian 07], partial least squares (PLS) [Sharma 11] and kernel PLS [Haj 12]. Both technics from [Marin-Jimenez 14] and [Murphy-Chutorian 07] estimate the pose angles independently, so several regression functions must be learned, one for each angle. Hence correlations between the head-pose angles cannot be taken into account during the learning. Another drawback of all kernel methods is that they require the design of a kernel function with its hyper-parameters, which must be either manually selected or properly estimated using non-convex optimization techniques.

PLS and kernel PLS proceed in two steps. First, both the input and the output are projected onto low-dimensional latent subspaces by maximizing the covariance between the projected input and the projected output. Second, a linear regression between these two latent subspaces is estimated. The performance of PLS depends on the relationship between the covariance matrices of input and output variables and on the eigen structure of the covariance of the input variable [Naik 00]. The advantage of this method is that it estimates a mixture of linear regressions directly from the input and output variables.

§ CNN

Convolutional neural network (CNN) architectures were also proposed in the recent past [Osadchy 07], [Ahn 14]. [Osadchy 07] considers a fixed image sub-window at all locations and scales. The network consists of $64,000$ weights and kernel coefficients that need to be estimated, and both face and non-face samples must be considered. Altogether, training the network with $52,000$ positives and $52,000$ negatives samples, involves non-linear optimization and takes 26 hours on a 2GHz Pentium 4. [Ahn 14] proposed a CNN architecture composed of four convolutional layers with max-pooling on the first two layers.

In their study, the activation function is the hyperbolic tangent which yields good convergence during the training phase. Small input RGB images (32×32 pixels) and small filters (5×5 pixels) are used in order to overcome the limitation of the training dataset. The network is trained using $13,500$ face patches extracted from the dataset. More recently, [Liu 16] suggested to simulate a dataset of head poses in order to train a CNN. Then they use the trained network to estimate head pose from real color images using the BIWI dataset [Fanelli 13]. They show that when trained using the BIWI dataset the CNN approach yields results similar to [Drouard 15] and that the accuracy is improved, by a factor of 2, when a large set of simulated images are used for training.

2.3 PROBABILISTIC PIECEWISE REGRESSION

In this section we will present the model used to estimate the head pose in an image. The model used is referred as GLLiM (Gaussian Locally Linear Mapping). It is a probabilistic model that learns a mapping between two spaces of different dimensions, a high-dimenssion one ($\in \mathbb{R}^D$) and a low-dimension one ($\in \mathbb{R}^L$) with $D \gg L$. The mapping is made by a mixture of local affine transformation. The pipeline for the learning and prediction is summarized in Figure (2.2).

2.3.1 INVERSE REGRESSION

In a classic regression formula between two random variables $\boldsymbol{X}$ and $\boldsymbol{Y}$, where $\boldsymbol{X} \in \mathbb{R}^L$ denotes the response variable (output) and $\boldsymbol{Y} \in \mathbb{R}^D$ the explanatory variable (input) with $D \gg L$, the objective is to learn the parameters of the regression from $\boldsymbol{Y}$ to $\boldsymbol{X}$ (the forward predictive regression). The specifity of the GLLiM model is that it inverses the roles of the input and output variables in the learning step, i.e. the high to low problem becomes a low to high problem. This dramastically drops the number of parameters and thus eases the training task. Then the parameters for the forward regression can be obtained without difficulty, this will be covered in the next subsection. The relation between $\boldsymbol{X}$ and $\boldsymbol{Y}$, possibly non-linear, is modeled using a mixture of *locally affine transformations*

$$\boldsymbol{Y} = \sum_{i=1}^{K} \mathbb{I}\{Z = i\} \left(\mathbf{A}_i \boldsymbol{X} + \boldsymbol{b}_i + \boldsymbol{e}_i\right), \tag{2.1}$$

where $\mathbb{I}$ is the identicator function and Z a discrete hidden variable used to specify the identity of the affine transformation between the two variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ such that $\boldsymbol{Y}$ is the image of $\boldsymbol{X}$ by the i^{th} affine transformation only if $\mathbb{I}\{Z = i\} = 1$. $\mathbf{A}_i \in \mathbb{R}^{D \times L}$ and $\boldsymbol{b}_i \in \mathbb{R}^D$ are the affine transformation parameters, $\boldsymbol{e}_i \in \mathbb{R}^D$ is a vector capturing the error due to the reconstruction using affine transformation and eventual noise in the observation. Figure 2.3 shows the relations between the three variables of the model.

The selection variable Z allows to rewrite Equation (2.1) in a probabilistic form as a

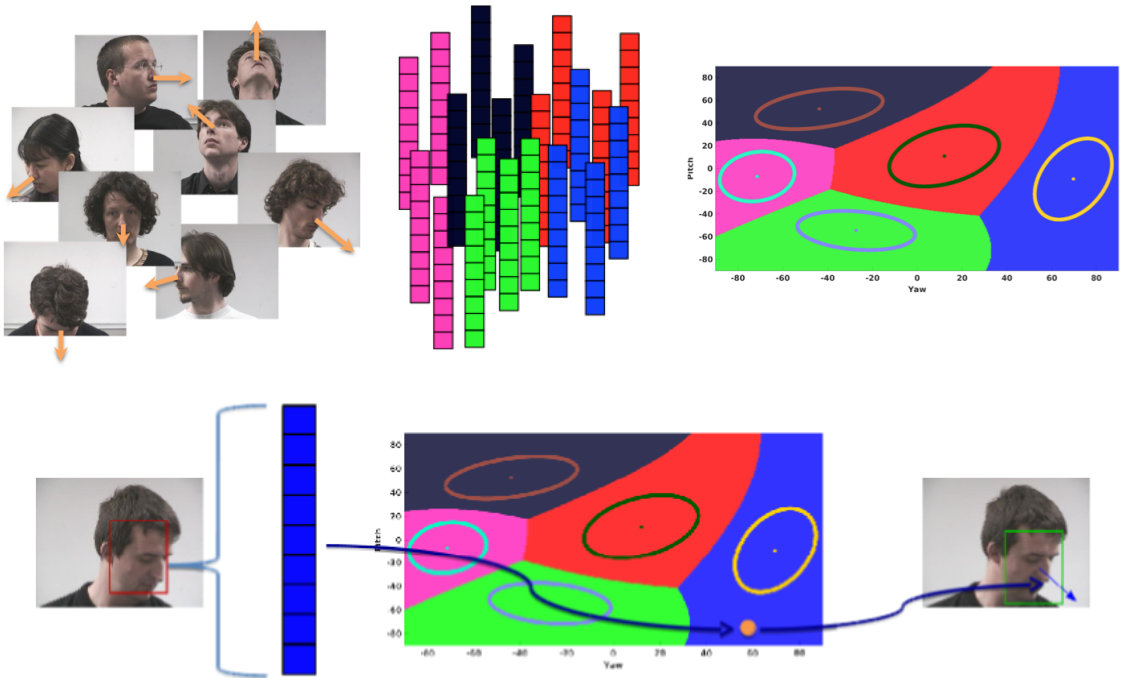

Figure 2.2: Pipeline of the proposed supervised head-pose estimation method. **Top:** the parameters of a mixture of linear regressions are learnt from faces annotated with their poses (left). The result of this learning is a simultaneous partitioning of both the high-dimensional input (high-dimensional feature vectors shown in the middle) and low-dimensional output (two-dimensional parameter space shown on the right), such that each region in this partition corresponds to an affine mapping between the input and the output. Moreover, the output is modeled by a Gaussian mixture and each region corresponds to a mixture component. This yields a predictive distribution that can then be used to predict an output from a test input. **Bottom:** A face detector is used to localize a bounding box (left, shown in red) from which a HOG descriptor, namely a high-dimensional feature vector, is extracted. Using the predictive distribution just mentioned, it is then possible to estimate the head-pose parameters (yaw and pitch in this example). Additionally, it is also possible to refine the bounding-box location such that the latter is optimally aligned with the face (right, shown in green).

distribution of $\boldsymbol{Y}$ conditioned by $\boldsymbol{X}$:

$$p(\boldsymbol{Y}=\boldsymbol{y}|\boldsymbol{X}=\boldsymbol{x};\boldsymbol{\theta}) = \sum_{i=1}^{K} p(\boldsymbol{Y}=\boldsymbol{y}|\boldsymbol{X}=\boldsymbol{x}, Z=i;\boldsymbol{\theta}) p(Z=i|\boldsymbol{X}=\boldsymbol{x};\boldsymbol{\theta}), \qquad (2.2)$$

where $\boldsymbol{\theta}$ denotes the model parameters and $\boldsymbol{y}$ and $\boldsymbol{x}$ denote realizations of $\boldsymbol{Y}$ and $\boldsymbol{X}$ respectively. Assuming that $\boldsymbol{e}_i$ is a zero-mean Gaussian variable with diagonal covariance matrix $\boldsymbol{\Sigma}_k \in \mathbb{R}^{D\times D}$ with diagonal entries $\sigma_{k1}, \ldots, \sigma_{kD}$, the conditional probability $p(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{X}=\boldsymbol{x}, Z=i;\boldsymbol{\theta})$ in Equation (2.2) can be expressed as Gaussian distribution of the following form (To simplify the visibility $\boldsymbol{Y}=\boldsymbol{y}$ and $\boldsymbol{X}=\boldsymbol{x}$ are replaced by $\boldsymbol{y}$ and $\boldsymbol{x}$):

$$p(\boldsymbol{y}|\boldsymbol{x}, Z=k;\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{y}; \mathbf{A}_k\boldsymbol{x} + \boldsymbol{b}_k, \boldsymbol{\Sigma}_k). \qquad (2.3)$$

If we further assume that $\boldsymbol{X}$ follows a mixture of Gaussians via the same assignment

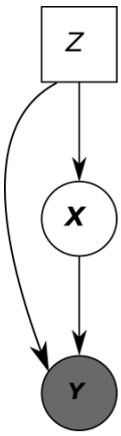

Figure 2.3: Graphical model representing the relations between the variable $\boldsymbol{Y}$, $\boldsymbol{X}$ and Z

variable $Z = k$, the following distributions can be defined:

$$p(\boldsymbol{x}|Z = k; \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{c}_k, \boldsymbol{\Gamma}_k), \tag{2.4}$$

$$p(Z = k; \boldsymbol{\theta}) = \pi_k, \tag{2.5}$$

where $\boldsymbol{c}_k \in \mathbb{R}^L$, $\boldsymbol{\Gamma}_k \in \mathbb{R}^{L \times L}$ and $\sum_{k=1}^{K} \pi_k = 1$. This assumption defines a partition of the lower dimensional space $\mathbb{R}^L$ into K regions $\mathcal{R}_i$, moreover using the same variable Z as in Equation (2.3) implies that if $\boldsymbol{x}$ lies in region $\mathcal{R}_i$ then the i^{th} affine transformation $(\mathbf{A}_i, \boldsymbol{b}_i)$ is going to be employed to express $\boldsymbol{y}$. Thus the model can be considered as a region mapping. This model is fully described by the parameter set:

$$\boldsymbol{\theta} = \{\boldsymbol{c}_k, \boldsymbol{\Gamma}_k, \pi_k, \mathbf{A}_k, \boldsymbol{b}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}. \tag{2.6}$$

Finally, when replacing $p(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{X} = \boldsymbol{x}, Z = i; \boldsymbol{\theta})$ and $p(Z = k|\boldsymbol{X} = \boldsymbol{x}; \boldsymbol{\theta})$ from Equation 2.2 by their true values using Equations (2.3), (2.4) and (2.5), the conditional distribution of $\boldsymbol{y}$ given $\boldsymbol{x}$ can now be expressed as follows:

$$p(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{i=1}^{K} \nu_i \mathcal{N}(\boldsymbol{y}; \mathbf{A}_i \boldsymbol{x} + \boldsymbol{b}_i, \boldsymbol{\Sigma}_i), \tag{2.7}$$

$$\text{with } \nu_i = \frac{\pi_i \mathcal{N}(\boldsymbol{x}; \boldsymbol{c}_i, \boldsymbol{\Gamma}_i)}{\sum_{i'=1}^{K} \pi_{i'} \mathcal{N}(\boldsymbol{x}; \boldsymbol{c}_{i'}, \boldsymbol{\Gamma}_{i'})}. \tag{2.8}$$

The parameters $\boldsymbol{\theta}$ are learned through an Expectation-Maximization (EM) algorithm shown in appendix A.

2.3.2 LATENT OUTPUT EXTENSION

In A compelling feature of the GLLiM model is the possibility to train the inverse regression in the presence of partially latent output $\boldsymbol{X}$, in this case we are referencing to

hybrid-GLLiM. While the high-dimensional variable $\boldsymbol{Y}$ remains unchanged, i.e. fully observed, the low-dimensional variable is a concatenation of an observed variable $\boldsymbol{T} \in \mathbb{R}^{L_t}$ and a latent variable $\boldsymbol{W} \in \mathbb{R}^{L_w}$, namely $\boldsymbol{X} = [\boldsymbol{T}; \boldsymbol{W}]$, where $[\Delta; \Delta]$ denotes vertical vector concatenation and with $L_t + L_w = L$. Hybrid-GLLiM can be seen as a latent-variable augmentation of standard regression. It can also be seen as a semi-supervised dimensionality reduction method since the unobserved low-dimensional variable $\boldsymbol{W}$ must be recovered from realizations of the observed variables $\boldsymbol{Y}$ and $\boldsymbol{T}$. The decomposition of $\boldsymbol{X}$ implies that some of the model parameters must be decomposed as well, namely $\boldsymbol{c}_k$, $\boldsymbol{\Gamma}_k$ and $\mathbf{A}_k$. Assuming the independence of $\boldsymbol{T}$ and $\boldsymbol{W}$ given Z we obtain:

$$\boldsymbol{c}_k = \begin{pmatrix} \boldsymbol{c}_k^t \\ \boldsymbol{c}_k^w \end{pmatrix}, \quad \boldsymbol{\Gamma}_k = \begin{pmatrix} \boldsymbol{\Gamma}_k^t & 0 \\ 0 & \boldsymbol{\Gamma}_k^w \end{pmatrix}, \quad \mathbf{A}_k = \begin{pmatrix} \mathbf{A}_k^t & \mathbf{A}_k^w \end{pmatrix}. \tag{2.9}$$

It follows that Equation (2.1) rewrites as:

$$\boldsymbol{Y} = \sum_{k=1}^{K} \mathbb{I}(Z = k)(\mathbf{A}_k^t \boldsymbol{T} + \mathbf{A}_k^w \boldsymbol{W} + \boldsymbol{b}_k + \boldsymbol{e}_k), \tag{2.10}$$

While the parameters to be estimated are the same, i.e. Equation (2.6) does not change, there are now two missing variables, $Z \in \{1 \dots K\}$ and $\boldsymbol{W} \in \mathbb{R}^{L_w}$, associated with the training data $(\boldsymbol{Y}, \boldsymbol{T})$. The means $\{\boldsymbol{c}_i^w\}_{i=1}^K$ and covariances $\{\boldsymbol{\Gamma}_i^w\}_{i=1}^K$ must be set in order to avoid non-identifiability issues. Indeed, changing their values corresponds to shifting and scaling the latent variable $\boldsymbol{W}$ which is compensated by changes in the parameters of the affine transformations $\{\mathbf{A}_i^w\}_{i=1}^K$ and $\{\boldsymbol{b}_i^w\}_{i=1}^K$. This identifiability problem is the same as the one encountered in latent variable models for dimension reduction and is always solved by fixing these parameters. Following [Ghahramani 96a] and [Tipping 99], the means and covariances are fixed to zero and to the identity matrix respectively: $\boldsymbol{c}_i^w = \mathbf{0}, \boldsymbol{\Gamma}_i^w = \mathbf{I}, \forall i \in \{1 \dots K\}$. Derivations to obtain the final conditional distribution are the same as in the case with no latent addition.

2.3.3 From inverse to forward

The desired high-dimensional (i.e. face descriptor) to low-dimensional (i.e. head pose) or forward predictive distribution can be obtained from the inverse predictive distribution once its parameters have been estimated. Using Bayes' inversion rule the forward predictive distribution is obtained and expressed as follows:

$$p(\boldsymbol{x}|\boldsymbol{y}; \boldsymbol{\theta}^*) = \sum_{k=1}^{K} \nu_k^* \mathcal{N}(\boldsymbol{x}; \mathbf{A}_k^* \boldsymbol{y} + \boldsymbol{b}_k^*, \boldsymbol{\Sigma}_k^*) \tag{2.11}$$

$$\text{with } \nu_k^* = \frac{\pi_k^* \mathcal{N}(\boldsymbol{y}; \boldsymbol{c}_k^*, \boldsymbol{\Gamma}_k^*)}{\sum_{j=1}^{K} \pi_j^* \mathcal{N}(\boldsymbol{y}; \boldsymbol{c}_j^*, \boldsymbol{\Gamma}_j^*)} \tag{2.12}$$

which is also a Gaussian mixture conditioned by the parameters $\boldsymbol{\theta}^*$:

$$\boldsymbol{\theta}^* = \{\boldsymbol{c}_k^*, \boldsymbol{\Gamma}_k^*, \pi_k^*, \mathbf{A}_k^*, \boldsymbol{b}_k^*, \boldsymbol{\Sigma}_k^*\}_{k=1}^K. \tag{2.13}$$

The forward predictive distribution follows also a Gaussian mixture, with the same selection variable Z, a notable feature of this model is that the parameters $\boldsymbol{\theta}^*$ can be expressed analytically from the parameters $\boldsymbol{\theta}$ as follows:

$$\boldsymbol{c}_k^* = \mathbf{A}_k \boldsymbol{c}_k + \boldsymbol{b}_k, \tag{2.14}$$
$$\boldsymbol{\Gamma}_k^* = \boldsymbol{\Sigma}_k + \mathbf{A}_k \boldsymbol{\Gamma}_k \mathbf{A}_k^\top, \tag{2.15}$$
$$\pi_k^* = \pi_k, \tag{2.16}$$
$$\mathbf{A}_k^* = \boldsymbol{\Sigma}_k^* \mathbf{A}_k^\top \boldsymbol{\Sigma}_k^{-1}, \tag{2.17}$$
$$\boldsymbol{b}_k^* = \boldsymbol{\Sigma}_k^* \left(\boldsymbol{\Gamma}_k^{-1} \boldsymbol{c}_k - \mathbf{A}_k^\top \boldsymbol{\Sigma}_k^{-1} \boldsymbol{b}_k\right), \tag{2.18}$$
$$\boldsymbol{\Sigma}_k^* = \left(\boldsymbol{\Gamma}_k^{-1} + \mathbf{A}_k^\top \boldsymbol{\Sigma}_k^{-1} \mathbf{A}_k\right)^{.-1} \tag{2.19}$$

The desired prediction $\hat{\boldsymbol{x}}$ of $\boldsymbol{X}$ given a realisation $\boldsymbol{y}$ of $\boldsymbol{Y}$ is obtained using Eq. 2.11. Two approaches can be used, the first one uses the expectation of Equation (2.11):

$$\hat{\boldsymbol{x}} = f(\hat{\boldsymbol{y}}) \quad \text{with:}$$
$$f(\boldsymbol{y}) = \mathbb{E}\left[\boldsymbol{x} | \boldsymbol{y}; \boldsymbol{\theta}^*\right] = \sum_{k=1}^{K} \nu_k^* \left(\mathbf{A}_k^* \boldsymbol{y} + \boldsymbol{b}_k^*\right). \tag{2.20}$$

The second one using the maximum a posteriori of $p(Z = i|\boldsymbol{y})$, selecting the most probable transformation given a new input $\boldsymbol{y}$.

In most cases these two approaches are very similar because optimally only one transformation should be selected. Nevertheless in the case where a new input $\boldsymbol{y}$ lies on the edge of two adjacent regions a soft combination of transformations might be preferable rather than a hard assignment.

2.4 EXPERIMENTAL VALIDATION

In this section we evaluate the performance of the method presented in previous section. Experiments are conducted on three publicly available datasets for head-pose estimation. In the first part, we conduct experiments to determine the optimal number of affine transformations. In the second part, we benchmark the method against some state of the art methods.

2.4.1 FACE REPRESENTATION

The proposed head-pose estimation method is implemented as follows. Faces were extracted using the Matlab computer vision toolbox implementation of the face detector of [Viola 01] as this method yields good face detections and localizations for a wide range of face orientations, including side views. The Matlab implementation of [Viola 01] offers three different trained classifiers for face detection: two of them for frontal-view detection and one for profile-view detection. These three classifiers yield different results for face detection in terms of bounding-box location and size. The results of face detection using these three classifiers are then combined for both training and testing of our method. For each face detection the associated bounding box is resized to patches of 64×64 pixels, and converted to a grey-level image. Then histogram equalization is then applied to the grey-level image. A HOG descriptor is extracted from this resized and histogram-equalized patch. To do so, a HOG pyramid (p-HOG) is built by stacking HOG descriptors at multiple resolutions. The following parameters are used to build p-HOG descriptors:

- Block resolution: 2×2 cells;
- Cell resolutions: 32×32, 16×16 and 8×8 pixels,
- Number of orientation bins: 8

Three HOG descriptors are computed, one for each cell resolution, which are then stacked to form a high-dimensional vector $\boldsymbol{y} \in \mathbb{R}^D$, with $D = 1888$.

2.4.2 THE DATASETS

The experiments are carried out with three publicly available datasets: the Prima dataset [Gourier 04], the Biwi Kinect dataset [Fanelli 13], and the McGill real-world face video dataset [Demirkus 13, Demirkus 15]:

- The **Prima head pose dataset** consists of 2790 images of 15 persons recorded twice. Pitch values lie in the interval $[-60°, 60°]$, and yaw values lie in the interval $[-90°, 90°]$ with a $15°$ step. Thus, there are 93 poses available for each person. Every recording was achieved with the same background. One interesting feature of this dataset is that the pose space is uniformly sampled. The dataset is annotated such that a face bounding box (manually annotated) and the corresponding yaw and pitch angle values are provided for each sample.

- The **Biwi Kinect head pose dataset** consists of video recordings of 20 people (16 men, 4 women, some of them recorded twice) using a Kinect camera. During the recordings, the participants freely move their head and the corresponding head angles lie in the intervals $[-60°, 60°]$ (pitch), $[-75°, 75°]$ (yaw), and $[-20°, 20°]$ (roll). Unlike the Prima dataset, the parameter space is not evenly sampled. The face centers (nose tips) were detected on each frame in the dataset, which allow to automatically extract a bounding box for each sample.
- The **McGill real-world face video dataset** consists of 60 videos (a single participant per video, 31 women and 29 men) recorded to study unconstrained head-pose estimation. The videos were recorded in different environments (both indoor and outdoor). This results in arbitrary illumination conditions and background clutter. Furthermore, the participants were completely free in their behaviors and movements. Yaw angles range in the interval $[-90°, 90°]$. Yaw values corresponding to each video frame are estimated using a two-step labelling procedure that provides the most likely angle as well as a degree of confidence. The labelling consists of showing images and possible angle values to human experts, i.e. [Demirkus 13].

2.4.3 THE PROTOCOL

Experiments were carried out using the leave-one-out evaluation protocol at the individual person level. This implies that all the images/frames associated with one participant were left aside and used for testing, while the remaining ones were used to train the models. Performance of the evaluation between the estimated angles with respect to the ground-truth one was done using the *mean absolute error* (MAE) and standard deviation (STD) over several tests, the following variations of the proposed method were experimented:

- *GLLiM_pose* learns and predicts the pose parameters;
- *hGLLiM_pose-d* learns and predicts the pose parameters as well as partially latent output, where the number d of latent variables varies between 1 and 4;
- *GLLiM_pose&bb* learns and predicts both pose angles and bounding-box shifts, and
- *hGLLiM_pose&bb-d* learns and predicts pose angles, bounding-box shifts and partially latent output.

An important aspect of any head-pose method is the way faces are detected in images. Manually annotated bounding boxes were used whenever they are available with the datasets. Otherwise, we used bounding boxes provided with a face detector, e.g. [Viola 01]. To evaluate the robustness in the presence of inaccurate face localization, random shifts, drawn from a Gaussian distribution, on the annotated face bounding-boxes were introduced, and we used these shifts in conjunction with *GLLiM_pose&bb* and with *hGLLiM_pose&bb-d* to learn the regression parameters and to predict the correct bounding-box location. In the case of the latter algorithms, the prediction is run iteratively. i.e. the algorithm extracts a HOG vector, predicts the pose and the shift, then it extracts a HOG

Algorithm 1 Iterative prediction

Require: Bounding-box location $\boldsymbol{u}$ and forward model parameters $\boldsymbol{\theta}^*$
procedure HEADPOSEESTIMATION($\boldsymbol{u}$,$\boldsymbol{\theta}^*$)
  repeat
    Build $\boldsymbol{y}$ from current bounding-box location $\boldsymbol{u}$
    Predict $\boldsymbol{x} = [\boldsymbol{x}_h; \boldsymbol{x}_b]$ from $\boldsymbol{y}$ using 2.20
    Update the bounding-box location $\boldsymbol{u} = \boldsymbol{u} + \boldsymbol{x}_b$
  until $\|\boldsymbol{x}_b\| \leq \epsilon$
  return head-pose $\boldsymbol{x}_h$ and bounding-box location $\boldsymbol{x}_b$
end procedure

vector from the shifted bounding box and predicts the pose and the shift, etc. This stops when the shift becomes very small. This scheme is explained in Algorithm 1.

The dimension of the output variable $\boldsymbol{x} \in \mathbb{R}^L$ depends on the number of pose parameters (up to three angles: yaw, pitch and roll), the bounding-box shift parameters (horizontal and vertical shifts) and the number of latent variables. Hence the output dimension may vary from $L = 1$ (one angle, no shift, no latent variable) to $L = 9$ (three angles, two shifts, four latent variables).

The joint estimation of the head-pose angles and bounding-box shift is achieved iteratively in the following way. The current bounding-box location, $\boldsymbol{u} \in \mathbb{R}^2$, and size $\boldsymbol{s} \in \mathbb{R}^2$, are used to build a feature vector $\boldsymbol{y}$ from which both a head pose $\boldsymbol{x}_h$ and a bounding-box shift $\boldsymbol{x}_b$ are predicted. The latter is then used to update the bounding-box location, to build an updated feature vector and to predict an updated head pose and a new bounding-box shift. This iterative prediction is described in detail in Algorithm 1.

2.4.4 RESULTS

§ Number of affine transformation

The number K of Gaussian components is an important parameter, as it corresponds, in the model to the number of affine mappings. Several experiments were carried out to evaluate the quality of the results obtained by our method as a function of the number of affine transformations in the mixture. To do that, we used the GLLiM_pose variant of our algorithm with three different face detection options: manual annotation (AFP), manual annotation perturbed with additive Gaussian noise (MNA), and automatic face detection (FD). These three versions of GLLiM_pose were trained with K varying from 1 to 100. In order to determine the optimal number of affine mappings, K, associated with GLLiM, we use two criteria, the Bayesian information criterion (BIC) which is a theoretic criterion generally used for model selection, and an experimental figure of merit based on the mean absolute error (MAE). Several models were learned for different values of K using the Prima dataset. We seek the model that yields low BIC and MAE scores. The BIC and MAE values are plotted as a function of K in Table 2.1 and in Fig. 2.4 and Fig. 2.5. These

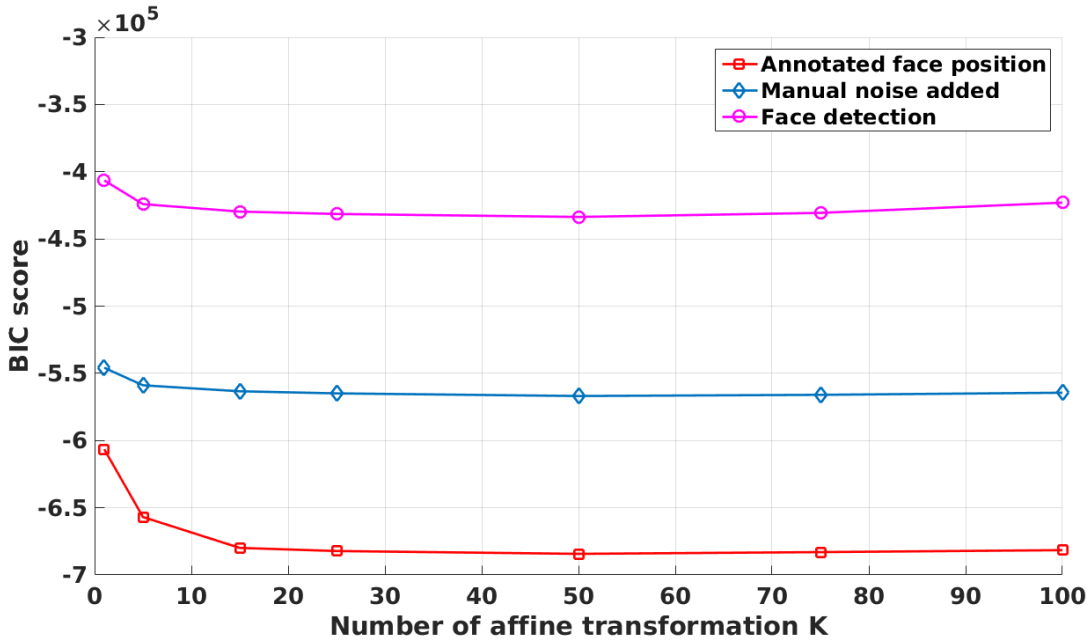

Figure 2.4: The Bayesian information criterion (BIC) as a function of the number of affine transformations in GLLiM. These experiments use the Prima dataset with the leave-one-out protocol.

Table 2.1: The BIC score for several models learned with different values of K using the Prima dataset. GLLiM_pose is used to learn each model with different input data (Fig. 2.4): annotated face position (AFP), adding manual noise to the face position (MNA) and using a face detector (FD), The optimal BIC scores are in bold.

Data	$K = 1$	$K = 5$	$K = 25$	$K = 50$	$K = 100$
AFP	-6.0608	-6.5845	-6.822	$\mathbf{-6.8429}$	-6.8173
MNA	-5.4554	-5.6018	-5.6491	$\mathbf{-5.6688}$	-5.6455
FD	-4.0596	-4.2602	-4.3144	$\mathbf{-4.3366}$	-4.2307

curves show the same behavior: as the number of affine mappings increases from $K = 1$ to approximatively $K = 30$, both the BIC and MAE scores decrease, then the curve slopes become almost horizontal. Both BIC and MAE reach the lowest score for $K = 50$. This behavior can be explained as follows. When $K < 5$ the model is not flexible enough to take into account the apparently non-linear mapping between HOG features and head-pose parameters. It can be observed from Fig. 2.5 that a large value for K increases the model accuracy. As expected, the computational complexity increases with K as well. Indeed, the number of model parameters is linear in the number of mixture components

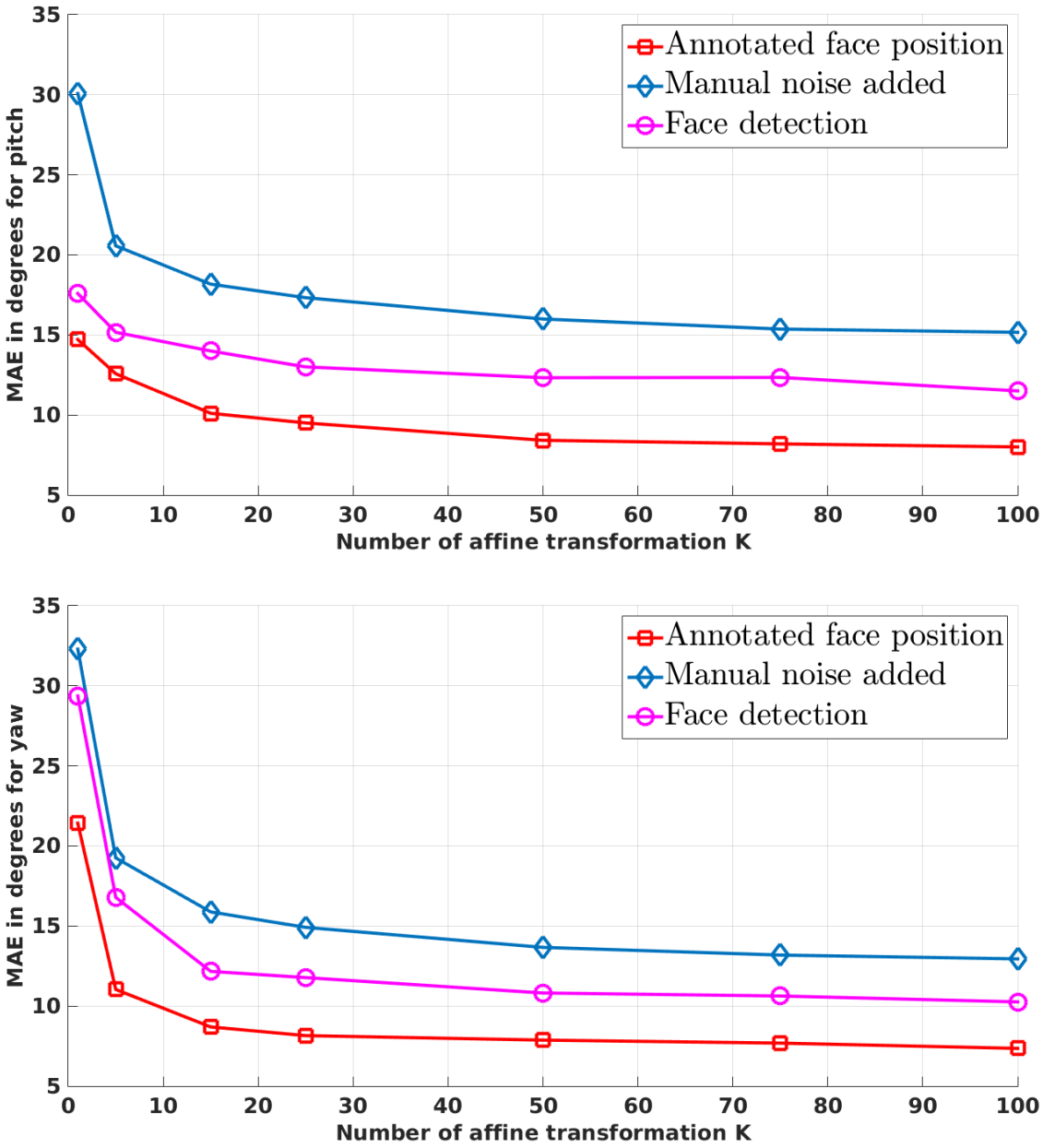

Figure 2.5: Mean absolute error (MAE) in degrees, for pitch (top) and yaw (bottom), as a function of the number of affine transformations in the mixture of linear regression model. GLLiM_pose is used to learn the model parameters independently for pitch and yaw. The three curves correspond to the following face detection cases: manual annotation (red curve), manual annotations with additive noise (blue), and automatic face detection (magenta). These experiments use the Prima dataset with the leave-one-out protocol.

and hence the size of the training dataset must be increased as well. It is well known that a large number of components in a mixture model presents the risk of overfitting. It is interesting to notice that BIC (derived from information theory) and MAE (based on

experiments with the data) yield the same optimal value, namely $K \approx 50$.

§ Comparison

The proposed algorithms (GLLiM_pose, hGLLiM_pose, ...) were compared with the following state-of-the-art head-pose estimation methods: the neural-network based methods of [Stiefelhagen 04], [Gourier 07] and of [Ahn 14] , the method of [Demirkus 12] based on dictionary learning, the graphical-model method of [Demirkus 14], the template based method of [Zhu 12], the supervised non-linear optimization method of [Xiong 13], the optimization method of [Ghiass 15], and the random-forest methods of [Fanelli 13] and of [Wang 13]. Additionally, the method was benchmarked with the following regression methods: support vector regression (SVR) [Smola 04], Gaussian process regression (GPR) [Rasmussen 06], and partial least squares (PLS) [Abdi 03]. We chose to compare our methods with these other methods for two reasons: they are widely known and commonly used regression methods. Some of these methods estimate only one parameter, i.e. the yaw angle [Demirkus 12, Demirkus 14, Zhu 12, Xiong 13], while the random-forest methods of [Fanelli 13], [Ghiass 15] and [Wang 13] use depth information available with the BIWI (Kinect) dataset.

Table 2.2, Table 2.3, and Table 2.4 show the results of head-pose estimation obtained with the Prima, Biwi Kinect, and McGill datasets, respectively. The † symbol indicates that the results are those reported by the authors while the ‡ symbol indicates that the results are obtained using either publicly available software packages or our own implementations. In the case of the Prima dataset, GLLiM_pose and hGLLiM_pose yield the best results. We note that hGLLiM_pose&bb variants of the algorithm (simultaneous prediction of pose, bounding-box shift and partially-latent output) increase the confidence (low STD). Table 2.3 shows the results obtained with the BIWI datasets. As already mentioned, [Fanelli 13] uses depth information and [Wang 13], [Ghiass 15] use of depth and color information. Overall, the proposed algorithms compare favorably with [Fanelli 13]. hGLLiM_pose-4 yields the best MAE for the roll angle, while [Ghiass 15] yields the best MAE for pitch and yaw, but with a high standard deviation. Our algorithms estimate the parameters with the highest confidence (lowest standard deviation). Table 2.4 shows the results obtained with the McGill dataset. The ground-truth yaw values in this dataset are obtained by human experts that must choose among a discrete set of 7 values. Clearly, this is not enough to properly train our algorithms. The method of [Demirkus 14] yields the best results in terms of RMSE while hGLLiM_pose-2 yields the best results in terms of MAE. Note that PLS yield the highest confidence in this case.

Table 2.2: Mean absolute error (MAE) and standard deviation (STD) (in degrees) obtained with various head-pose methods, regression methods, and our methods using the Prima dataset. This dataset contains manually annotated bounding boxes of faces and the corresponding pitch and yaw angles. In order to test the robustness we simulated shifted bounding boxes. The best results are in bold.

	Manually annotated bounding boxes				Manual annotation + simulated shifts			
	Pitch		Yaw		Pitch		Yaw	
Method	MAE	STD	MAE	STD	MAE	STD	MAE	STD
Stiefelhagen [Stiefelhagen 04]‡	9.7	-	9.5	-	-	-	-	-
Gourier et al. [Gourier 07]‡	12.1	-	7.3	-	-	-	-	-
GPR [Rasmussen 06]†	11.94	10.19	15.04	12.24	19.96	16.58	23.69	18.16
PLS [Abdi 03]†	12.25	9.73	13.38	10.8	17.77	14.47	17.34	13.94
SVR [Smola 04]†	11.25	9.42	12.82	10.99	17.09	14.81	17.27	14.09
GLLiM_pose	***8.41***	***10.65***	7.87	8.08	15.99	16.69	13.66	14.78
hGLLiM_pose-2	**8.47**	**10.35**	**7.93**	**7.9**	12.64	14.49	11.51	11.37
hGLLiM_pose-4	8.5	10.8	***7.85***	***7.98***	**12.03**	13.92	**10.78**	9.77
GLLiM_pose&bb	-	-	-	-	13.13	13.65	11.3	10.55
hGLLiM_pose&bb-2	-	-	-	-	12.52	**12.44**	***11.04***	9.7
hGLLiM_pose&bb-4	-	-	-	-	***12.12***	***12.85***	11.27	**9.53**

Table 2.3: The mean absolute error (MAE) and standard deviation (STD), expressed in degrees, obtained with various head-pose estimation methods, regression methods, and our methods using the Biwi Kinect dataset. This dataset contains annotated bounding boxes of faces and the corresponding pitch, yaw, and roll angle values. In order to test the robustness we simulated shifted bounding boxes both for training and for testing. The best results are in bold. Note that [Fanelli 13] uses depth data only and [Wang 13] and [Ghiass 15] use both color and depth information. Papers using depth data are marked with a *.

	Manually annotated bounding boxes						Manual annotation + simulated shifts					
	Pitch		Yaw		Roll		Pitch		Yaw		Roll	
Method	MAE	STD	MAE	STD	MAE	STD	MAE	STD	MAE	STD	MAE	STD
Ghiass et al. [Ghiass 15]‡*	**0.1**	6.7	**0.2**	8.7	**0.3**	9.3	-	-	-	-	-	-
Fanelli et al. [Fanelli 13]‡*	3.8	6.5	3.5	6.8	5.4	6.0	-	-	-	-	-	-
Wang et al. [Wang 13]‡*	8.5	11.1	8.8	14.3	7.4	10.8	-	-	-	-	-	-
GPR [Rasmussen 06]†	9.64	8.85	7.72	7.17	6.01	6.29	10.77	9.45	9.06	8.33	6.54	6.72
PLS [Abdi 03]†	7.87	6.73	7.35	6.06	6.11	5.9	10.32	8.64	8.67	7.7	6.69	6.74
SVR [Smola 04]†	7.77	6.85	6.98	6.26	5.14	5.96	10.82	9.22	9.14	8.32	6.26	7.16
GLLiM_pose	5.77	5.77	4.48	**4.33**	4.71	5.31	11.33	11.58	10.2	11.34	7.76	8.02
hGLLiM_pose-2	5.57	5.48	***4.33***	4.68	4.37	5.09	9.04	9.13	7.65	8.3	6.3	6.75
hGLLiM_pose-4	***5.43***	**5.44**	***4.24***	5.37	***4.13***	**4.86**	8.45	8.41	6.93	7.72	6.12	6.8
GLLiM_pose&bb	-	-	-	-	-	-	8.49	8.79	6.86	7.3	6.57	6.95
hGLLiM_pose&bb-2	-	-	-	-	-	-	***7.81***	***7.68***	***6.41***	***7.19***	***5.75***	***6.68***
hGLLiM_pose&bb-4	-	-	-	-	-	-	**7.65**	**8.0**	**6.06**	**6.91**	**5.62**	**6.35**

Table 2.4: Root mean square error (RMSE), mean absolute error (MAE) and standard deviation (STD) (in degrees) obtained with various head-pose methods, regression methods, and our methods using the McGill Real-World dataset. This dataset contains annotated yaw angles. Bounding boxes are located with a face detector. The best results are in bold.

	Bounding boxes based on face detection		
	Yaw		
Method	RMSE	MAE	STD
Demirkus et al. [Demirkus 12]‡	> 40	-	-
Xiong and De la Torre [Xiong 13]‡	29.81	-	-
Zhu and Ramanan [Zhu 12]‡	35.70	-	-
Demirkus et al. [Demirkus 14]‡	**12.41**	-	-
GPR [Rasmussen 06]†	23.18	16.22	***16.71***
PLS [Abdi 03]†	22.46	15.56	**16.2**
SVR [Smola 04]†	21.13	15.25	18.43
GLLiM_pose	26.62	13.1	23.17
hGLLiM_pose-2	24.0	**11.99**	20.79
hGLLiM_pose-4	24.25	***12.01***	21.06

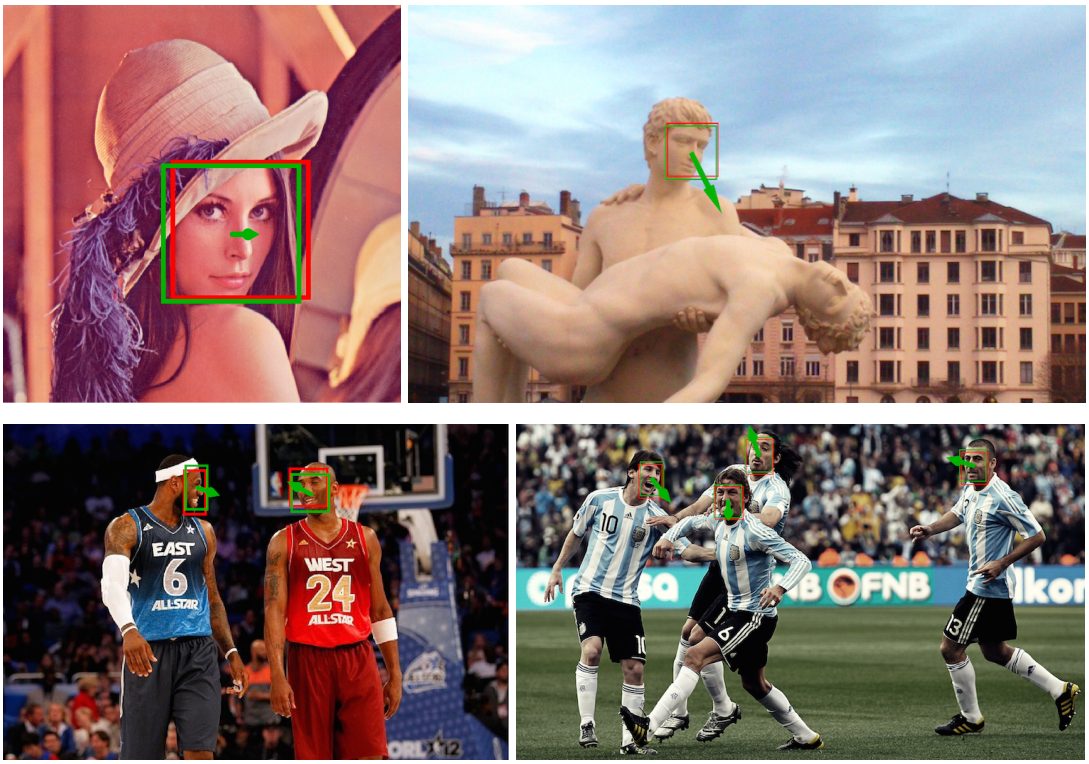

Figure 2.6: Examples of simultaneous estimation of head-pose angles and of bounding-box shifts. The initial bounding box (found with an automatic face detector) is shown in red. The estimated bounding box is shown in green.

2.5 CONCLUSION

In this chapter, we introduced a new method to estimate the head pose from the bounding box of a face. Inspired by [Deleforge 15], we used the idea of mapping two spaces of different dimension through a combination of local affine transformations and applied it for head-pose estimation. The method model the mapping between the low-dimensional head-pose space through a Gaussian mixture model. The missing data variable, of the mixture of Gaussian model, selects the optimal transformation to use given an observation. To facilitate the learning of the parameters, the method learns the inverse regression from pose to features, and then followed by Bayesian inversion to obtain the forward regression parameters. Misalignment of the face detection leads to wrong estimation of the head-pose, we improve the robustness of the estimation by estimating both the head pose and the misalignment offset simultaneously. and concatenating them in the object state vector. We designed a recursive algorithm using this combination, that estimates the head pose and refines the face bounding box until convergence of the latter. Some results can be visualized on Figure 2.6. The algorithm to solve head-pose estimation presented has not been tuned for this particular application. The formulas are generic and can easily be applied, with minor modifications, to other high-dimensional to low-dimensional mapping problems, e.g. [Deleforge 14] used the method for sound source localization in a image. In the present model, each estimation is based on single image, this leads to estimation on a sequence that is not smooth, by incorporating information from past, the estimation on sequence could be ameliorated. In the next chapter we introduce a temporal dynamic on the state variables to solve the tracking problem. Two publications came out of this work, [Drouard 17b] and [Drouard 15] which received the best student paper award ($2nd$ place) at the IEEE International Conference on Image Processing (ICIP) in 2015.

CHAPTER 3

SWITCHING DYNAMICAL MODEL FOR HEAD-POSE ESTIMATION

This chapter addresses the problem of head-pose tracking. The work presented inside is an extension of the model presented in Chapter 2. It combines the regression model with a dynamic model in the same probabilistic framework. This is achieved by embedding the Gaussian mixture of linear inverse-regression model into a dynamic Bayesian model, this combination is called a switching linear temporal model. The idea through this model is to build a robust temporal estimation, using past to improve present prediction. The resulting is a smoother transition between two consecutive estimations with less variation and to overcome possible bad estimation due to noisy observation in a frame. We formally derive the equations of the proposed switching linear regression model. Unfortunatelly the tracking becomes quickly intractable due to an exponentially increment of the number of components in the final mixture model with time. Thus we propose two approximations that are both identifiable and computationally tractable, two EM procedures were designed to estimate the model parameters with closed-form expressions. Experiments and comparisons were carried out with other methods on two publicly available datasets.

3.1 HEAD POSE AND PROBABILISTIC TRACKING

The proposed method combines high-dimensional to low-dimensional mixture of linear regressions with a switching state-space model. In practice we adopt two approximations of the obtained temporal model, that yield closed-form expressions for the estimation of the tracked parameters. Hence, it is more efficient than sampling techniques which are often used in conjunction with generative tracking methods.

3.1.1 RELATED WORK

Head-pose tracking is not new and has been an actively investigated topic; head-pose estimation and tracking methods were surveyed [Murphy-Chutorian 09]. Many approaches rely on extracting facial landmarks, then tracking these landmarks over the image sequence to finally estimate a rigid transformation between consecutive images, e.g. [Gee 96, Uřičář 12], or between consecutive image pairs, e.g. [Yang 02]. Similarly, [Maurer 96] builds a face graph based on the landmarks and tracks this graph over the image sequence. Another landmark-based approach [Yao 01] consists of using a 3D model of a generic face that embeds model-centered coordinates of facial landmarks, e.g. nose tip, eyes, lip corners, etc. The model is first fitted to the face detected in the first image and then fitted to the subsequent faces by tracking the landmarks. These methods heavily rely on landmark detection and tracking as well as on the robust estimation of the 2D-landmark-to-3D-landmark rigid transformation, i.e. the pose parameters. Therefore these methods are limited to frontal views of faces, because the landmarks are partially or totally occluded in side views of faces. Moreover, they track the facial landmarks instead of the pose parameters, hence they do not yield smooth pose trajectories. The advantage of the proposed method is that it relies neither on facial landmark detection nor on landmark tracking. The proposed method, once trained based on pairs of feature descriptors and pose parameters, can deal with side views of faces, unlike landmark-based methods. Head-pose tracking was also addressed using sampling methods based on particle filters, which allow to sample the temporal predictive distribution e.g. [Ba 04]. A principled way of combining a latent-variable temporal filter with the observed data is an important issue. In [Tu 06] it is proposed to extract a high-dimensional feature vector from a face and then to apply PCA to reduce its dimensionality. This assumes that the high-dimensional to low-dimensional mapping is linear (which may not be the case) and it does not guarantee that the PCA output contains pose information. Particle filtering can also be combined with a 3D deformable model and with facial landmarks, e.g. [Dornaika 04, Taheri 13]. As already outlined, landmark extraction is not always possible. The advantage of the proposed method over these particle-filter trackers is both theoretical and methodological: the feature-space to parameter-space mapping is combined with a dynamic model, and the estimation of the model parameters yields closed-form EM procedures.

Switching state space models have also been used to solve tracking problems. For example, [Ghahramani 96b], [Oh 05] and [Kooij 12] show that the use of switching linear models helps tracking. In [Pavlovic 00] switching models are applied for tracking people in videos in order to obtain motion-capture data, and three different approaches for inferring the parameters are compared, namely the Viterbi algorithm, variational inference, and the generalized pseudo Bayesian algorithm of order 2 (GPB2). The reported results obtained with these three approaches are quite similar. Viterbi has the lowest complexity, GPB2 yields the smoothest parameter trajectories, while the variational inference achieves a good compromise between low complexity and smooth trajectories.

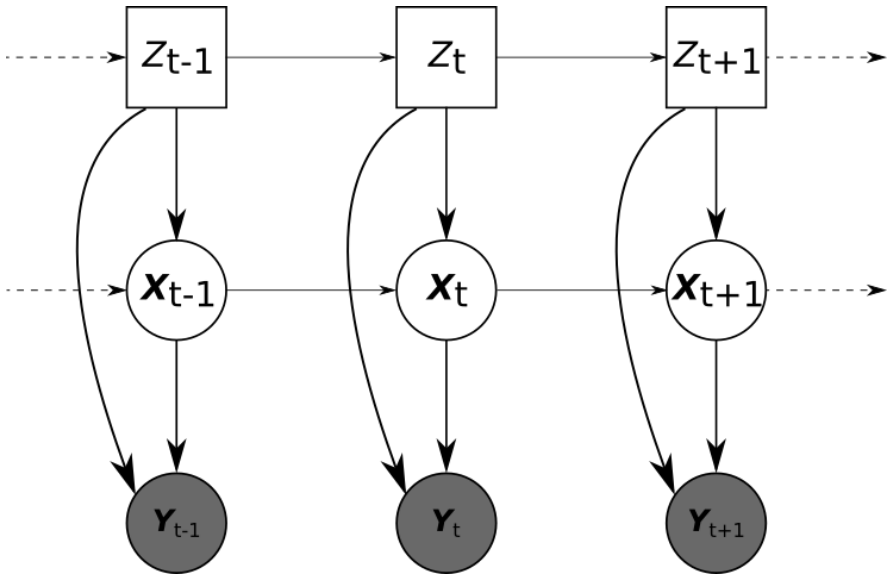

Figure 3.1: The temporal graphical model, temporal extension of Figure 2.3 with a dynamic on the latent variable $\boldsymbol{X}$ and Z

3.2 TEMPORAL MIXTURE OF LINEAR REGRESSIONS

We wanted to have a global framework that could use the estimation model presented in Chapter 2 but that would also incorporate information from the past. In this context, we focus on probabilistic models for tracking and especially switching temporal models for tracking. Using as observation model, the model previously presented, we incorporate a new temporal equation to model the temporal relation between two consecutive object state values. We wish to extend the model of Chapter 2 to a temporal model for tracking the head pose over time. The main difference between the probabilistic regression model presented in Chapter 2 Section 2.3 and the proposed temporal model is that the conditional distribution $p(\boldsymbol{x}|\boldsymbol{y})$ is replaced with $p\left(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}\right)$, where t is the time index. The proposed graphical model is shown on Figure 3.1, where Z_t is the discrete latent variable associated with the Gaussian mixture of linear regression, $\boldsymbol{X}_t$ and $\boldsymbol{Y}_t$ are the latent head pose and the observed high-dimensional feature vector at t, respectively. Using marginalization rule to make appear the latent variables $\boldsymbol{X}$ and Z at time $t-1$, the new posterior conditional distribution is expressed as follows:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \sum_{j=1}^{K}\sum_{i=1}^{K}\int_{\boldsymbol{X}_{t-1}} p(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}, Z_t = j, Z_{t-1} = i|\boldsymbol{y}_{1:t})d\boldsymbol{x}_{t-1}. \tag{3.1}$$

Under the Markovian assumption and using the conditional independencies associated with the proposed graphical model of Figure 3.1, the term insides the integral of Equation

(3.1) can be decomposed as follows:

$$\begin{aligned} p(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}, & Z_t = j, Z_{t-1} = i, \boldsymbol{y}_t | \boldsymbol{y}_{1:t-1}) \\ &= p\left(\boldsymbol{y}_t | \boldsymbol{x}_t, Z_t = j\right) p\left(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, Z_t = j\right) \\ &\quad p\left(Z_t = j | Z_{t-1} = i\right) p\left(\boldsymbol{x}_{t-1} | Z_{t-1} = i, \boldsymbol{y}_{1:t-1}\right) p\left(Z_{t-1} = i | \boldsymbol{y}_{1:t-1}\right). \end{aligned} \tag{3.2}$$

Among the right hand side probabilities of this equation, $p\left(\boldsymbol{y}_t | \boldsymbol{x}_t, Z_t = j\right)$ is the observation model introduced in Equation (2.3), the other distributions are defined as follows:

$$p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, Z_t = j) = \mathcal{N}(\boldsymbol{x}_t; \mathbf{C}_j \boldsymbol{x}_{t-1}, \mathbf{Q}_j), \tag{3.3}$$

$$p(\boldsymbol{x}_{t-1} | Z_{t-1} = i, \boldsymbol{y}_{1:t-1}) = \mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\eta}^i_{t-1}, \mathbf{V}_i). \tag{3.4}$$

These last 2 equations are addition due to the dynamic model on $\boldsymbol{x}$. Where $\mathbf{C}_j$ is the dynamic transition matrix between $\boldsymbol{x}_t$, $\boldsymbol{x}_{t-1}$ and $\mathbf{Q}_j$ its associated covariance, $\boldsymbol{\eta}^i_{t-1}$ and $\mathbf{V}_i$ the estimation mean and associated covariance at $t-1$ given $Z_{t-1} = i$. The other terms of the right side of Equation (3.2) replace Equations (2.4) and (2.5) and are defined as follows:

$$p(Z_t = j | Z_{t-1} = i) = \tau_{ij}, \tag{3.5}$$

$$p(Z_{t-1} = i | \boldsymbol{y}_{t-1}) = \nu^i_{t-1}. \tag{3.6}$$

$p(Z_t = j | Z_{t-1} = i)$ is called the switching probability. $p(\boldsymbol{x}_{t-1} | \boldsymbol{y}_{1:t-1})$ is defined as a mixture of Gaussian with Z_{t-1} as the selection discrete latent variable and with K components. The parameters related to the temporal model will be jointly denoted by $\boldsymbol{\phi}$:

$$\boldsymbol{\phi} = \{\mathbf{C}_j, \mathbf{Q}_j, \tau_{ij}, i, j = 1 \dots K\}. \tag{3.7}$$

By substituting Equations (2.3), (3.3) and (3.4) into Equation (3.2), the final posterior distribution is now:

$$\begin{aligned} p(\boldsymbol{x}_t | \boldsymbol{y}_{1:t}) = & \sum_{j=1}^{K} \sum_{i=1}^{K} \tau_{ij} \nu^i_{t-1} \mathcal{N}(\boldsymbol{y}_t; \mathbf{A}_j \boldsymbol{x}_t + \boldsymbol{b}_j, \boldsymbol{\Sigma}_j) \\ & \times \int_{\boldsymbol{X}_{t-1}} \mathcal{N}(\boldsymbol{x}_t; \mathbf{C}_j \boldsymbol{x}_{t-1}, \mathbf{Q}_j) \mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\eta}^i_{t-1}, \mathbf{V}_i) d\boldsymbol{x}_{t-1}. \end{aligned} \tag{3.8}$$

The product of Gaussian distributions inside the integral can be rewritten as a product of two Gaussian distributions as follows (proof can be found in Chapter 2 of [Bishop 07], Equations 2.113-2.117):

$$\begin{aligned} & \mathcal{N}(\boldsymbol{x}_t; \mathbf{C}_j \boldsymbol{x}_{t-1}, \mathbf{Q}_j) \mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\eta}^i_{t-1}, \mathbf{V}_i) \\ & \qquad = \mathcal{N}(\boldsymbol{x}_t; \mathbf{C}_j \boldsymbol{\eta}^i_{t-1}, \mathbf{Q}_j + \mathbf{C}_j \mathbf{V}_i \mathbf{C}_j^\top) \mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{h}, \mathbf{H}), \end{aligned} \tag{3.9}$$

where:

$$\begin{aligned} \mathbf{H} &= \left(\mathbf{C}_j^\top \mathbf{Q}_j^{-1} \mathbf{C}_j + \mathbf{V}_i^{-1}\right)^{-1}, \\ \boldsymbol{h} &= \mathbf{H}\left(\mathbf{Q}_j^{-1} \mathbf{C}_j \boldsymbol{x}_t + \mathbf{V}_i^{-1} \boldsymbol{\eta}^i_{t-1}\right). \end{aligned}$$

The first Gaussian in Equation (3.9) is called the predictive distribution, it does not depend on $\boldsymbol{x}_{t-1}$ anymore, thus it can be put outside of the integral. The second one is a Gaussian distribution over $\boldsymbol{x}_{t-1}$ and thus integrating it over $\boldsymbol{x}_{t-1}$ will make it disappear from the posterior distribution:

$$\begin{aligned} &p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) \\ &\quad= \sum_{j=1}^{K}\sum_{i=1}^{K} \tau_{ij}\nu_{t-1}^{i}\mathcal{N}(\boldsymbol{y}_t; \mathbf{A}_j\boldsymbol{x}_t + \boldsymbol{b}_j, \boldsymbol{\Sigma}_j)\mathcal{N}(\boldsymbol{x}_t; \mathbf{C}_j\boldsymbol{\eta}_{t-1}^{i}, \mathbf{Q}_j + \mathbf{C}_j\mathbf{V}_i\mathbf{C}_j^{\top}). \end{aligned} \tag{3.10}$$

The multiplication of the two remaining Gaussian distributions will result on a Gaussian distribution on $\boldsymbol{x}_t$ and a residual term that is also a Gaussian distribution but that does not depend on $\boldsymbol{x}_t$ this term can be understood as the distance between the true observation and the predicted observation given $\boldsymbol{x}_{t-1}$ if the value of Z_t is j, it is a way of determining what is the best value of Z_t given the current observation and the estimated value of $\boldsymbol{x}_{t-1}$.

$$\begin{aligned} &\mathcal{N}(\boldsymbol{y}_t; \mathbf{A}_j\boldsymbol{x}_t + \boldsymbol{b}_j, \boldsymbol{\Sigma}_j)\mathcal{N}(\boldsymbol{x}_t; \mathbf{C}_j\boldsymbol{\eta}_{t-1}^{i}, \mathbf{Q}_j + \mathbf{C}_j\mathbf{V}_i\mathbf{C}_j^{\top}) \\ &\qquad= \mathcal{N}(\boldsymbol{d}_{t|t-1}^{ij}; 0, \mathbf{S}_{t|t-1}^{ij})\mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_{t|t-1}^{ij}, \mathbf{W}_{t|t-1}^{ij}). \end{aligned} \tag{3.11}$$

The parameters of the Gaussian distributions are expressed as a function of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$:

$$\mathbf{W}_{t|t-1}^{ij} = \left(\boldsymbol{\Sigma}_j^{*-1} + \mathbf{P}_{t-1}^{ij}\right)^{-1}, \tag{3.12}$$

$$\boldsymbol{\mu}_{t|t-1}^{ij} = \mathbf{W}_{t|t-1}^{ij}\left(\boldsymbol{\Sigma}_j^{*-1}\left(\mathbf{A}_j^{*}\boldsymbol{y}_t + \boldsymbol{b}_j^{*}\right) + \mathbf{P}_{t-1}^{ij}\mathbf{C}_j\boldsymbol{\eta}_{t-1}^{j}\right), \tag{3.13}$$

$$\boldsymbol{d}_{t|t-1}^{ij} = \mathbf{y}_t - \mathbf{A}_j(\mathbf{C}_j\boldsymbol{\eta}_{t-1}^{i}) - \boldsymbol{b}_j, \tag{3.14}$$

$$\mathbf{S}_{t|t-1}^{ij} = \boldsymbol{\Sigma}_j + \mathbf{A}_j(\mathbf{Q}_j + \mathbf{C}_j\mathbf{V}_{t-1}^{i}\mathbf{C}_j^{\top})\mathbf{A}_j^{\top}. \tag{3.15}$$

with

$$\mathbf{P}_{t-1}^{ij} = \left(\mathbf{Q}_j + \mathbf{C}_j\mathbf{V}_{t-1}^{i}\mathbf{C}_j^{\top}\right)^{-1}, \tag{3.16}$$

$$\mathbf{A}_j^{*} = \boldsymbol{\Sigma}_j^{*}\mathbf{A}_j^{\top}\boldsymbol{\Sigma}_j^{-1}, \tag{3.17}$$

$$\boldsymbol{b}_j^{*} = -\mathbf{A}_j^{*}\boldsymbol{b}_j, \tag{3.18}$$

$$\boldsymbol{\Sigma}_j^{*} = (\mathbf{A}_j^{\top}\boldsymbol{\Sigma}_j^{-1}\mathbf{A}_j)^{-1}. \tag{3.19}$$

One interesting consequence of replacing Equations (2.4) and (2.5) with Equations (3.3) to (3.6) is that the formulas in Equations (2.17) to (2.19) are now simplified (Equations (3.17) to (3.19)) and thus the parameters set $\boldsymbol{\theta}$ can be reduced to $\boldsymbol{\theta}_r = \{\mathbf{A}_j, \boldsymbol{b}_j, \boldsymbol{\Sigma}_j\}_{j=1}^{K}$.

Finally the posterior distribution can be fully expressed as:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \sum_{i=1}^{K}\sum_{j=1}^{K} \pi_{t|t-1}^{ij}\mathcal{N}(\boldsymbol{x}_t|\boldsymbol{\mu}_{t|t-1}^{ij}, \mathbf{W}_{t|t-1}^{ij}), \tag{3.20}$$

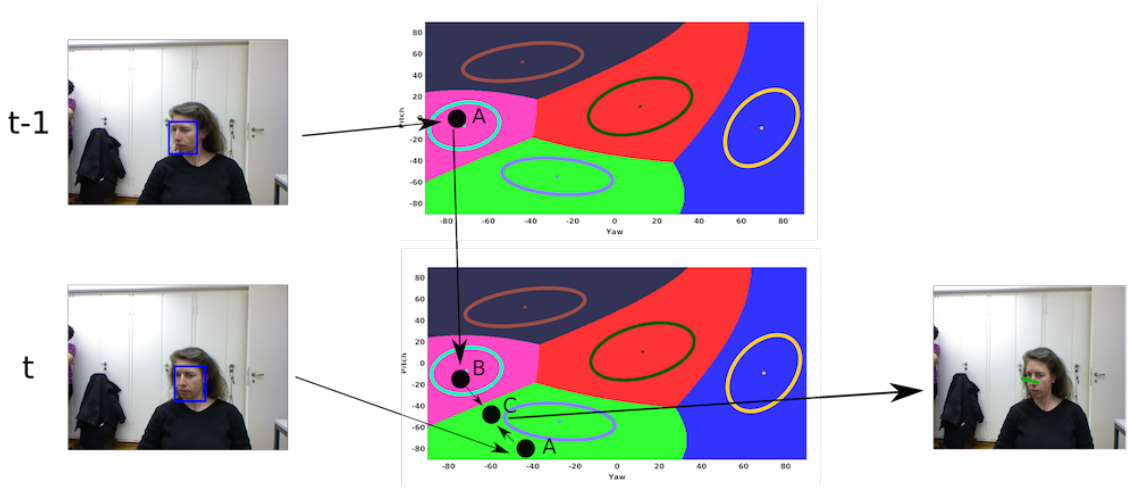

Figure 3.2: The method starts by learning a mixture of linear regression that allows the prediction of a head-pose from a feature vector obtained from the bounding box of a face. Hence, Equation 2.11 (2) is applied at $t-1$ (top) and at t (bottom) and head poses are thus predicted, they are denoted A on the figure. Notice that, because of various perturbations in the data and of inherent flaws in face detection, the two predictions use two different affine transformations and hence they are associated with two different Gaussian components in the mixture, i.e *magenta* and *green* on the figure. The proposed dynamic model combines the temporal prediction of the filter from $t-1$ to t, denoted B on the figure, with the pose predicted at t, to yield a filtered pose estimate, denoted C on the figure. The mixture of linear regression is plugged in the SKF model in a principled way.

where:

$$\pi_{t|t-1}^{ij} = \rho_{t-1}^{i} \tau_{ij} \mathcal{N}(\boldsymbol{d}_{t|t-1}^{ij}; 0, \mathbf{S}_{t|t-1}^{ij}), \tag{3.21}$$

$$\boldsymbol{\psi}_{t|t-1} = \{\pi_{t|t-1}^{ij}, \boldsymbol{\mu}_{t|t-1}^{ij}, \mathbf{W}_{t|t-1}^{ij}, i, j = 1 \dots K\}. \tag{3.22}$$

The mean defined in Equation (3.13) can be seen as a "weighted" linear combination of the dynamical prediction $\mathbf{C}_j \eta_{t-1}^{j}$ and of the prediction based on observation $\mathbf{A}_j^* \boldsymbol{y}_t + \boldsymbol{b}_j^*$, where the "weights" are covariance matrices. Thus the confidence related to the covariance matrices defines the weights of the dynamical prediction and the observation prediction in the final estimation. Eq. (3.12) is the associated covariance matrix, which is the inverse of the sum of the precision matrix of the temporal prediction $\mathbf{P}_{t-1}^{ij}$ and precision matrix $\boldsymbol{\Sigma}_j^{*-1}$ of the observation $\boldsymbol{y}_t$. The GMM proportions in Equation (3.21) are defined as a product between three terms: the proportions of the i^{th} components at $t-1$, ρ_{t-1}^{i}, the switching filter transition probabilities τ_{ij}, and $\mathcal{N}(\boldsymbol{d}_{t|t-1}^{ij}; 0, \mathbf{S}_{t|t-1}^{ij})$. The pipeline is summarized in Figure 3.2

3.3 HANDLING THE COMPONENTS GROWTH

The underlying problem with switching temporal models is the exponentially growth of the number of components in the mixture of Gaussian of the posterior distribution. Equations (3.4) and (3.6) defines a mixture of K Gaussian a time $t-1$ but the posterior at time t, $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t};\boldsymbol{\psi}_{t|t-1})$, is a mixture of K^2 Gaussians, Equation (3.20). This growth of the number of components is due to the dynamic model on Z_t, Equation (3.5).

The number of Gaussian components in the posterior distribution increases exponentially at each time step, making the model intractable with time. Computing the parameters for each Gaussian will be more and more time consuming and also the storage of these parameters can be difficult in some cases. For example in the case where the algorithm is running embedded on a robot with few memory space and limited computing capacity it becomes necessary to contain the number of components in the mixture to avoid the exponential explosion of this number.

Two approaches were invertigated to overcome this issue. First using the *generalized pseudo Bayesian algorithm of order 2* (GPB2) and using a variational approximation to estimate the posterior distribution and fixing the number of components in the mixture for each time step. These two approaches gives different output, according to [Pavlovic 00], the GPB2 gives a smoother output but the variational approximation has lower computational time and complexity.

3.3.1 APPROXIMATION USING THE GPB2 ALGORITHM

The generalized Pseudo Bayesian algorithm of order 2 (GPB2) is an algorithm that reduces components in a mixture model by merging moments. The order of the algorithm (in this case step 2) will determine how to merge the components. When the order is 2 all the components that diverge in the history from 2 steps are merged, i.e. the components with the same value for Z_t will be merged together and thus the resulting will be a distribution with K components. Using the mixture reduction scheme explained in [Salmond 09], the parameters of the K-component GMM $\boldsymbol{\lambda}_t$ can now be evaluated from the parameters of the K^2-component GMM $\boldsymbol{\psi}_{t|t-1}$, with the following formulas:

$$\boldsymbol{\eta}_t^j = \sum_{i=1}^{K} \tilde{\pi}_{t|t-1}^{ij} \boldsymbol{\mu}_{t|t-1}^{ij}, \tag{3.23}$$

$$\mathbf{V}_t^j = \sum_{i=1}^{K} \tilde{\pi}_{t|t-1}^{ij} \big(\mathbf{W}_{t|t-1}^{ij} + (\boldsymbol{\mu}_{t|t-1}^{ij} - \boldsymbol{\eta}_t^j)(\boldsymbol{\mu}_{t|t-1}^{ij} - \boldsymbol{\eta}_t^j)^\top\big), \tag{3.24}$$

$$\boldsymbol{\rho}_t^j = \sum_{i=1}^{K} \pi_{t|t-1}^{ij}, \tag{3.25}$$

$$\text{with } \tilde{\pi}_{t|t-1}^{ij} = \pi_{t|t-1}^{ij} / \sum_{k=1}^{k} \pi_{t|t-1}^{kj}. \tag{3.26}$$

Algorithm 2 Tracking with GPB2 algorithm

Require: posterior distribution at time $t-1$
1: **procedure** HEADPOSETRACKING($\boldsymbol{y}_{1:T}$, $\boldsymbol{\theta}$, $\boldsymbol{\phi}$)
2: $\boldsymbol{\lambda}_1 \leftarrow$ HeadPoseEstimation($\boldsymbol{y}_1$,$\boldsymbol{\theta}$)
3: t = 2
4: **repeat**
5: Compute parameters of Equation (3.20) $\boldsymbol{\psi}_{t|t-1}$ from $\boldsymbol{y}_t$, $\boldsymbol{\lambda}_{t-1}$ using $\boldsymbol{\theta}_r$, $\boldsymbol{\phi}$
6: Compute $\boldsymbol{\lambda}_t$ using GPB2 from $\boldsymbol{\psi}_{t|t-1}$
7: **until** t = T
8: **return** head-pose estimation $\boldsymbol{\eta}_{1:T}$
9: **end procedure**

The approximate posterior distribution is now:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t};\boldsymbol{\lambda}_t) \approx \sum_{j=1}^{K} \rho_t^j \mathcal{N}(\boldsymbol{x}_t;\boldsymbol{\eta}_t^j,\mathbf{V}_t^j). \tag{3.27}$$

with the parameters set:

$$\boldsymbol{\lambda}_t = \{\rho_t^j, \boldsymbol{\eta}_t^j, \mathbf{V}_t^j, j = 1 \dots K\}. \tag{3.28}$$

ρ_t^j defines the posterior distribution of $Z_t = j$, $p(Z_t = j|\boldsymbol{y}_{1:t})$ and $\mathcal{N}(\boldsymbol{x}_t;\boldsymbol{\eta}_t^j,\mathbf{V}_t^j)$ corresponds to $p(\boldsymbol{x}_t|Z_t = j,\boldsymbol{y}_{1:t})$. The K-component GMM approximation of Equation (3.20) of the K^2-component GMM (3.27) guarantees the computational tractability of the temporal model.

3.3.2 VARIATIONAL APPROXIMATION

The GPB2 algorithm allows to handle the growing number of components in the mixture though it still requires at each step to compute the parameters for K^2 components and this is cumbersome. To avoid the computation of so many Gaussian parameters we proposed a variational approximation whose goal is to find a simpler distribution to the original one that is computationally tractable using the Kullback–Leibler divergence:

$$D_{KL}\left(p(x)||q(x)\right) = \int_{-\infty}^{+\infty} p\left(x\right) \log\left(\frac{p(x)}{q(x)}\right) dx, \tag{3.29}$$

with $p(x)$ is the original distribution and $q(x)$ the approximated one. Variational approximation allows us to control the posterior distribution. To reduce the complexity with respect to the approximation using the GPB2 algorithm we decided to break the time dependency and the dependency between the two latent variables. Thus we define the following variational approximation for the joint posterior distribution:

$$p(\boldsymbol{x}_{1:t}, Z_{1:t}|\boldsymbol{y}_{1:t}) \approx \prod_{t=1}^{t} q(\boldsymbol{x}_t) \prod_{t=1}^{t} q(Z_t), \tag{3.30}$$

where $q(\boldsymbol{x}_t)$ and $q(Z_t)$ are the variational aproximation of the posterior probability of $\boldsymbol{x}_t$ and Z_t. The optimal values for $q^*(Z_t)$ and $q^*(\boldsymbol{x}_t)$ that minimize the Kullback-Leibler divergence have the following expression:

$$\log q^*(Z_t) = \mathbb{E}_{q(\boldsymbol{x}_{1:t}, Z_{1:t}\setminus Z_t)}\left[\log p(\boldsymbol{x}_{1:t}, Z_{1:t}|\boldsymbol{y}_{1:t})\right], \tag{3.31}$$

$$\log q^*(\boldsymbol{x}_t) = \mathbb{E}_{q(\boldsymbol{x}_{1:t}\setminus \boldsymbol{x}_t, Z_{1:t})}\left[\log p(\boldsymbol{x}_{1:t}, Z_{1:t}|\boldsymbol{y}_{1:t})\right]. \tag{3.32}$$

To compute Equations (3.31) and (3.32) we need first to express the log likelihood:

$$\log p(\boldsymbol{x}_{1:t}, Z_{1:t}, \boldsymbol{y}_{1:t}) \approx \sum_{t=2}^{t}\sum_{j=1}^{K}\alpha_{t,j}\Bigg[\log p(\boldsymbol{y}_t|\boldsymbol{x}_t, Z_t=j) + \log p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, Z_t=j) + \sum_{i=1}^{K}\alpha_{t-1,i}\log p(Z_t=j|Z_{t-1}=i)\Bigg], \tag{3.33}$$

where $\alpha_{t,j}=1$ if $Z_t=j$, 0 otherwise.

§ Estimation of the posterior distribution of Z_t

The optimal log posterior distribution of Z_t can now be obtained by replacing the loglikelihood by its true value (Equation (3.33)) in Equation (3.31):

$$\begin{aligned}\log q^*(Z_t) \approx& \mathbb{E}_{q(\boldsymbol{x}_{1:t}, Z_{1:t}\setminus Z_t)}\left[\log p(\boldsymbol{x}_{1:t}, Z_{1:t-1}, Z_t=j, \boldsymbol{y}_{1:t})\right],\\ \approx& \sum_{j=1}^{K}\alpha_{t,j}\Bigg[\mathbb{E}_{q(\boldsymbol{x}_t)}\left[\log p(\boldsymbol{y}_t|\boldsymbol{x}_t, Z_t=j)\right] + \mathbb{E}_{q(\boldsymbol{x}_t)q(\boldsymbol{x}_{t-1})}\left[\log p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, Z_t=j)\right]\\ &+ \sum_{i=1}^{K}\alpha_{t-1,i}\mathbb{E}_{q(Z_{t-1})}\left[\log p(Z_t|Z_{t-1})\right]\Bigg].\end{aligned} \tag{3.34}$$

By replacing each element of Equation (3.34) by their true value we obtain:

$$\begin{aligned}q^*(Z_t) \propto& \prod_{j=1}^{K}\Bigg[\mathcal{N}(\boldsymbol{y}_t; \mathbf{A}_j\mathbb{E}[\boldsymbol{x}_t] + \boldsymbol{b}_j, \boldsymbol{\Sigma}_j)\exp\left(-\frac{1}{2}\mathrm{Tr}\left[\mathbf{A}_j^\top\boldsymbol{\Sigma}_j^{-1}\mathbf{A}_j\mathrm{Cov}(\boldsymbol{x}_t)\right]\right)\\ &\times\mathcal{N}\left(\mathbb{E}[\boldsymbol{x}_t]; \mathbf{C}_j\mathbb{E}[\boldsymbol{x}_{t-1}], \mathbf{Q}_j\right)\exp\left(-\frac{1}{2}\mathrm{Tr}\left[\mathbf{C}_j^\top\mathbf{Q}_j^{-1}\mathbf{C}_j\mathrm{Cov}(\boldsymbol{x}_{t-1})\right]\right)\\ &\times\exp\left(-\frac{1}{2}\mathrm{T}\left[\mathbf{Q}_j^{-1}\mathrm{Cov}(\boldsymbol{x}_t)\right]\right)\prod_{j=1}^{K}a_{ij}^{\mathbb{E}[\alpha_{t-1,i}]}\Bigg]^{\alpha_{t,j}},\end{aligned} \tag{3.35}$$

Tr is the expression of the trace of a matrix. Using the results of Equation (3.35) we can express $q^*(Z_t=j)$ as follows:

$$q^*(Z_t=j) \approx \frac{q^*(Z_t|\alpha_{t,j}=1)}{\sum_{i=1}^{K}q^*(Z_t|\alpha_{t,i}=1)}. \tag{3.36}$$

§ Estimation of the posterior distribution of $\boldsymbol{x}_t$

With the GPB2 algorithm, the posterior probability of $\boldsymbol{x}_t$ is defined as a mixture of Gaussians with the Z_t, but with the variational approximation, because the dependency between $\boldsymbol{x}_t$ and Z_t no longer exists a posteriori, $q(\boldsymbol{x}_t)$ is expressed as a single Gaussian distribution:

$$q^*\left(\boldsymbol{x}_t\right) \propto \mathcal{N}\left(\boldsymbol{X}_t; \boldsymbol{\eta}_t, \mathbf{V}_t\right) \tag{3.37}$$

The derivation of Equation (3.32) follows the same principle as the one of Equation (3.31), we want:

$$\log q^*\left(\boldsymbol{x}_t\right) = \mathbb{E}_{q(\boldsymbol{x}_{1:t}\setminus\boldsymbol{x}_t, Z_{1:t})}\left[\log p\left(\boldsymbol{x}_{1:t}, Z_{1:t}|\boldsymbol{y}_{1:t}\right)\right] \tag{3.38}$$

$$\approx \sum_{j=1}^{K} \mathbb{E}_{q(Z_t)}\left[\alpha_{t,j}\right]\left[\log p(\boldsymbol{y}_t|\boldsymbol{x}_t, Z_t = j) + \mathbb{E}_{q(\boldsymbol{x}_{t-1})}\left[\log p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, Z_t = j)\right]\right], \tag{3.39}$$

by deriving this probability we obtained the desired Gaussian distribution with the parameters expressed as follows:

$$\mathbf{V}_t = \left[\sum_{j=1}^{K} \mathbb{E}[\alpha_{t,j}]\left(\mathbf{A}_j^\top \boldsymbol{\Sigma}_j^{-1}\mathbf{A}_j + \mathbf{Q}_j^{-1}\right)\right]^{-1}, \tag{3.40}$$

$$\boldsymbol{\eta}_t = \mathbf{V}_t\left[\sum_{j=1}^{K} \mathbb{E}[\alpha_{t,j}]\left(\mathbf{A}_j^\top \boldsymbol{\Sigma}_j^{-1}(\boldsymbol{y}_t - \boldsymbol{b}_t) + \mathbf{Q}_j^{-1}\mathbf{C}_j\mathbb{E}[\boldsymbol{x}_{t-1}]\right)\right]. \tag{3.41}$$

We define $\mathbb{E}[\boldsymbol{x}_t] = \boldsymbol{\eta}_t$, $\mathrm{Cov}(\boldsymbol{x}_t) = \mathbf{V}_t$ and $\mathbb{E}[\alpha_{t,j}] = q^*(Z_t = j)$. It is important to notice that $q(\boldsymbol{x}_t)$ and $q(Z_t)$ depend on the future, indeed $\mathbb{E}[\boldsymbol{x}_{t+1}]$ and $\mathbb{E}[\alpha_{t+1,i}]$ appear in the final expressions, for an online estimation one can remove the part with future in $q(\boldsymbol{x}_t)$ and $q(Z_t)$.

The complete derivation of the variational distribution of $\boldsymbol{x}_t$ and Z_t can be found in Appendix B.

3.3.3 DISCUSSION

With respect to the GPB2 approximation, our variational approximation strongly decreases the time required to compute the posterior distribution, of $\boldsymbol{x}_t$, parameters. While the approximation using the GPB2 algorithm requires to compute K^2 Gaussian parameters Equations (3.12), (3.13) and (3.21) before reducing them to K Gaussian ones Equations (3.62-3.64), the proposed variational approximation only requires the computation of the parameters of a single Gaussian Equations (3.40) and (3.41), thus reducing considerably the time required to compute the posterior distributions.

3.4 PARAMETERS ESTIMATION

The parameters of the model $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are learned separately. First $\boldsymbol{\theta}$ are learned using the algorithm described in Chapter 2 and Appendix A. We use an EM procedure to learn the parameters $\boldsymbol{\phi}$. We present the two steps of the algorithm for both approximations.

3.4.1 E STEP USING GBP2 APPROXIMATION

The E-step will compute the posterior distributions of the latent variables of the model namely $\boldsymbol{x}_t$ and Z_t. For the learning phase we are not limited, we can use the full sequence of observations ($\boldsymbol{y}_{1:T}$, $1 \leq t \leq T$), using all the observations will result in giving more information, thus more accurate estimation. We first need to derive the posterior distribution using the full sequence of observation $p\left(\boldsymbol{x}_t|\boldsymbol{y}_{1:T}\right)$, we will refer it as the smoothing distribution. Using Equations (13.32) and (13.33) from [Bishop 07] the smoothing distribution can be expressed as:

$$p\left(\boldsymbol{x}_t|\boldsymbol{y}_{1:T}\right) = \sum_{j=1}^{K} p\left(\boldsymbol{x}_t, Z_t = j|\boldsymbol{y}_{1:T}\right), \tag{3.42}$$

$$= \sum_{j=1}^{K} p(\boldsymbol{x}_t, Z_t = j|\boldsymbol{y}_{1:t}) p(\boldsymbol{y}_{t+1:T}|\boldsymbol{x}_t, Z_t = j), \tag{3.43}$$

where $p\left(\boldsymbol{x}_t, Z_t = j|\boldsymbol{y}_{1:T}\right)$ is obtained from Equation (3.27) and is called the forward distribution, $p(\boldsymbol{y}_{t+1:T}|\boldsymbol{x}_t, Z_t = j)$ is called the backward distribution and is derived using the same principles as the forward one:

$$p(\boldsymbol{y}_{t+1:T}|\boldsymbol{x}_t, Z_t = j) = \sum_{i=1}^{K} \int_{\boldsymbol{x}_{t+1}} p(\boldsymbol{y}_{t+1:T}, \boldsymbol{x}_{t+1}, Z_{t+1} = i|\boldsymbol{x}_t, Z_t = j) d\boldsymbol{x}_{t+1} \tag{3.44}$$

$$= \sum_{i=1}^{K} p(Z_{t+1} = i|Z_t = j) \int_{\boldsymbol{x}_{t+1}} p(\boldsymbol{y}_{t+1}|\boldsymbol{x}_{t+1}, Z_{t+1} = i) \times p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, Z_{t+1} = i) p(\boldsymbol{y}_{t+2:T}|\boldsymbol{x}_{t+1}, Z_{t+1} = i) d\boldsymbol{x}_{t+1} \tag{3.45}$$

where $p(\boldsymbol{y}_{t+2:T}|\boldsymbol{x}_{t+1}, Z_{t+1} = i) = \mathcal{N}(\boldsymbol{x}_{t+1}; \boldsymbol{\eta}^{i,b}_{t+1}, \mathbf{V}^{i,b}_{t+1})$, the other distributions have been defined in Section 3.2. We wish to express the 3 distributions inside the integral differently in order to isolate $\boldsymbol{x}_{t+1}$ and thus remove the integral. We rewrote the product of Gaussian as follow, first:

$$p(\boldsymbol{y}_{t+1}|\boldsymbol{x}_{t+1}, Z_{t+1} = i) p(\boldsymbol{y}_{t+2:T}|\boldsymbol{x}_{t+1}, Z_{t+1} = i)$$

$$= \mathcal{N}(\boldsymbol{x}_{t+1}; \boldsymbol{\eta}^{i,b}_{t+1}, \mathbf{V}^{i,b}_{t+1}) \mathcal{N}\left(\boldsymbol{y}_{t+1}; \mathbf{A}_i \boldsymbol{x}_{t+1} + \boldsymbol{b}_i, \boldsymbol{\Sigma}_i\right) \tag{3.46}$$

$$\propto \mathcal{N}\left(\boldsymbol{y}_{t+1}; \mathbf{A}_i \boldsymbol{\eta}^{i,b}_{t+1} + \boldsymbol{b}_i, \left(\mathbf{A}_j \mathbf{V}^{i,b}_{t+1} \mathbf{A}_j^\top + \boldsymbol{\Sigma}_i^{-1}\right)\right) \mathcal{N}\left(\boldsymbol{x}_{t+1}; \boldsymbol{m}^i_{t+1}, \mathbf{M}_{t+1}\right) \tag{3.47}$$

with

$$\mathbf{M}_{t+1}^{i} = \left({\mathbf{V}_{t+1}^{ib}}^{-1} + \mathbf{A}_i^\top \mathbf{\Sigma}_i^{-1} \mathbf{A}_i \right)^{-1} \tag{3.48}$$

$$\boldsymbol{m}_{t+1}^{i} = \mathbf{M}_{t+1} \left({\mathbf{V}_{t+1}^{ib}}^{-1} \boldsymbol{\eta}_{t+1}^{ib} + \mathbf{A}_i^\top \mathbf{\Sigma}_i^{-1} \left(\boldsymbol{y}_{t+1} - \boldsymbol{b}_i \right) \right) \tag{3.49}$$

The dynamical distribution of $\boldsymbol{x}_{t+1}$ can be rewritten as follows:

$$p(\boldsymbol{x}_{t+1} | \boldsymbol{x}_t, Z_{t+1} = i) = \mathcal{N}\left(\boldsymbol{x}_{t+1}; \mathbf{C}_i \boldsymbol{x}_t, \mathbf{Q}_i\right) \tag{3.50}$$

$$\propto \mathcal{N}\left(\boldsymbol{x}_t; \mathbf{C}_i^* \boldsymbol{x}_{t+1}, \mathbf{Q}_i^*\right). \tag{3.51}$$

with:

$$\mathbf{Q}_i^* = \left(\mathbf{C}_i^\top \mathbf{Q}_i \mathbf{C}_i \right)^{-1}, \tag{3.52}$$

$$\mathbf{C}_i^* = \mathbf{Q}_i^* \mathbf{C}_i^\top \mathbf{Q}_i^{-1}. \tag{3.53}$$

Finally the last product:

$$\begin{aligned} &\mathcal{N}\left(\boldsymbol{x}_{t+1}; \boldsymbol{m}_{t+1}^{i}, \mathbf{M}_{t+1}^{i}\right) \mathcal{N}\left(\boldsymbol{x}_t; \mathbf{C}_i^* \boldsymbol{x}_{t+1}, \mathbf{Q}_i^*\right) \\ &\propto \mathcal{N}\left(\boldsymbol{x}_t; \mathbf{C}_i^* \boldsymbol{m}_{t+1}^{ib}, \left(\mathbf{Q}_i^* + \mathbf{C}_i^* \mathbf{M}_{t+1}^{i} {\mathbf{C}_i^*}^\top \right)\right) \mathcal{N}\left(\boldsymbol{x}_{t+1}; \boldsymbol{n}_{t+1}^{i}, \mathbf{N}_{t+1}^{i}\right). \end{aligned} \tag{3.54}$$

The last Gaussian distribution will disappear in Equation (3.45) thanks to the integration over $\boldsymbol{x}_{t+1}$, thus the parameters $\boldsymbol{n}_{t+1}^{i}, \mathbf{N}_{t+1}^{i}$ don't need to be expressed. Now the backward distribution becomes:

$$\begin{aligned} p(\boldsymbol{y}_{t+1:T} | \boldsymbol{x}_t, Z_t = j) \propto \sum_{i=1}^{K} \tau_{ij} &\mathcal{N}\left(\boldsymbol{y}_{t+1}; \mathbf{A}_i \boldsymbol{\eta}_{t+1}^{i,b} + \boldsymbol{b}_i, \left(\mathbf{A}_j \mathbf{V}_{t+1}^{i,b} \mathbf{A}_j^\top + \mathbf{\Sigma}_i^{-1} \right)\right) \\ &\times \mathcal{N}\left(\boldsymbol{x}_t; \mathbf{C}_i^* \boldsymbol{m}_{t+1}^{ib}, \left(\mathbf{Q}_i^* + \mathbf{C}_i^* \mathbf{M}_{t+1}^{i} {\mathbf{C}_i^*}^\top \right)\right). \end{aligned} \tag{3.55}$$

By replacing the smoothing distribution in Equation (3.45) by its true value (Equation (3.55)), the final posterior distribution is:

$$\begin{aligned} p\left(\boldsymbol{x}_t | \boldsymbol{y}_{1:T}\right) \propto &\sum_{j=1}^{K} \sum_{i=1}^{K} \rho_t^j \tau_{ij} \mathcal{N}\left(\boldsymbol{y}_{t+1}; \mathbf{A}_i \boldsymbol{\eta}_{t+1}^{i,b} + \boldsymbol{b}_i, \left(\mathbf{A}_j \mathbf{V}_{t+1}^{i,b} \mathbf{A}_j^\top + \mathbf{\Sigma}_i^{-1} \right)\right) \\ &\times \mathcal{N}\left(\boldsymbol{x}_t; \boldsymbol{\eta}_t^j, \mathbf{V}_t^j\right) \mathcal{N}\left(\boldsymbol{x}_t; \mathbf{C}_i^* \boldsymbol{m}_{t+1}^{ib}, \left(\mathbf{Q}_i^* + \mathbf{C}_i^* \mathbf{M}_{t+1}^{i} {\mathbf{C}_i^*}^\top \right)\right) \end{aligned} \tag{3.56}$$

$$\propto \sum_{j=1}^{K} \sum_{i=1}^{K} \pi_{t|t+1}^{ij} \mathcal{N}\left(\boldsymbol{x}_t; \boldsymbol{\mu}_{t|t+1}^{ij}, \mathbf{W}_{t|t+1}^{ij}\right), \tag{3.57}$$

where the parameters of the mixture are defined as follows:

$$\pi_{t|t+1}^{ij} = \rho_t^j \tau_{ij} \mathcal{N}\left(\boldsymbol{y}_{t+1}; \mathbf{A}_i \boldsymbol{\eta}_{t+1}^{i,b} + \boldsymbol{b}_i, \left(\mathbf{A}_j \mathbf{V}_{t+1}^{i,b} \mathbf{A}_j^\top + \mathbf{\Sigma}_i^{-1} \right)\right) \tag{3.58}$$

$$\mathbf{W}_{t|t+1}^{ij} = \left({\mathbf{V}_t^j}^{-1} + \left(\mathbf{Q}_i^* + \mathbf{C}_i^* \mathbf{M}_{t+1}^{i} {\mathbf{C}_i^*}^\top \right)^{-1} \right)^{-1} \tag{3.59}$$

$$\boldsymbol{\mu}_{t|t+1}^{ij} = \mathbf{W}_{t|t+1}^{ij} \left({\mathbf{V}_t^j}^{-1} \boldsymbol{\eta}_t^j + \left(\mathbf{Q}_i^* + \mathbf{C}_i^* \mathbf{M}_{t+1}^{i} {\mathbf{C}_i^*}^\top \right)^{-1} \mathbf{C}_i^* \boldsymbol{m}_{t+1}^{i} \right) \tag{3.60}$$

Like in the posterior distribution is also a mixture of K^2 Gaussian, we apply the GBP2 algorithm to reduce to mixture of K Gaussian distributions to obtain:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:T}) \approx \sum_{j=1}^{K} \rho_t^{jb} \mathcal{N}\left(\boldsymbol{x}_t; \boldsymbol{\eta}_t^{jb}, \mathbf{V}_t^{jb}\right), \tag{3.61}$$

where:

$$\boldsymbol{\eta}_t^{jb} = \sum_{i=1}^{K} \tilde{\pi}_{t|t+1}^{ij} \boldsymbol{\mu}_{t|t+1}^{ij}, \tag{3.62}$$

$$\mathbf{V}_t^{j} = \sum_{i=1}^{K} \tilde{\pi}_{t|t-1}^{ij} \big(\mathbf{W}_{t|t+1}^{ij} + (\boldsymbol{\mu}_{t|t+1}^{ij} - \boldsymbol{\eta}_t^{jb})(\boldsymbol{\mu}_{t|t+1}^{ij} - \boldsymbol{\eta}_t^{jb})^\top\big), \tag{3.63}$$

$$\rho_t^{jb} = \sum_{i=1}^{K} \pi_{t|t+1}^{ij}, \tag{3.64}$$

$$\text{with } \tilde{\pi}_{t|t+1}^{ij} = \pi_{t|t+1}^{ij} / \sum_{k=1}^{k} \pi_{t|t+1}^{kj}. \tag{3.65}$$

3.4.2 E STEP USING VARIATIONAL APPROXIMATION

The smoothing distribution for $\boldsymbol{x}_t$ and Z_t for the variational approximation is obtained by solving the following equations:

$$\log q^*(Z_t) = \mathbb{E}_{q(\boldsymbol{X}, Z \setminus Z_t)} \left[\log p(\boldsymbol{x}_{1:T}, Z_{1:T}|\boldsymbol{y}_{1:T})\right], \tag{3.66}$$

$$\log q^*(\boldsymbol{x}_t) = \mathbb{E}_{q(\boldsymbol{X}, Z \setminus \boldsymbol{x}_t)} \left[\log p(\boldsymbol{x}_{1:T}, Z_{1:T}|\boldsymbol{y}_{1:T})\right]. \tag{3.67}$$

The derivations follow the same step as in the filtering case, and we obtain the following expression for Z_t:

$$\begin{aligned} q^*(Z_t) \propto \prod_{j=1}^{K} \Bigg[& \mathcal{N}(\boldsymbol{y}_t; \mathbf{A}_j \mathbb{E}[\boldsymbol{x}_t] + \boldsymbol{b}_j, \boldsymbol{\Sigma}_j) \exp\left(-\frac{1}{2}\mathrm{Tr}\left[\mathbf{A}_j^\top \boldsymbol{\Sigma}_j^{-1} \mathbf{A}_j \mathrm{Cov}(\boldsymbol{x}_t)\right]\right) \\ & \times \mathcal{N}\left(\mathbb{E}[\boldsymbol{x}_t]; \mathbf{C}_j \mathbb{E}[\boldsymbol{x}_{t-1}], \mathbf{Q}_j\right) \exp\left(-\frac{1}{2}\mathrm{Tr}\left[\mathbf{C}_j^\top \mathbf{Q}_j^{-1} \mathbf{C}_j \mathrm{Cov}(\boldsymbol{x}_{t-1})\right]\right) \\ & \times \exp\left(-\frac{1}{2}\mathrm{Tr}\left[\mathbf{Q}_j^{-1} \mathrm{Cov}(\boldsymbol{x}_t)\right]\right) \prod_{j=1}^{K} a_{ij}^{\mathbb{E}[\alpha_{t-1,i}]} \prod_{j=1}^{K} a_{ij}^{\mathbb{E}[\alpha_{t+1,i}]} \Bigg]^{\alpha_{t,j}}. \end{aligned} \tag{3.68}$$

Which allows us to express $q^*(Z_t = j)$ as follows:

$$q^*(Z_t = j) \approx \frac{q^*(Z_t|\alpha_{t,j} = 1)}{\sum_{i=1}^{K} q^*(Z_t|\alpha_{t,i} = 1)}. \tag{3.69}$$

And for $\boldsymbol{x}_t$ we obtain for the parameters of the Gaussian the following expressions:

$$\mathbf{V}_t = \left[\sum_{j=1}^{K} \mathbb{E}[\alpha_{t,j}] \left(\mathbf{A}_j^\top \boldsymbol{\Sigma}_j^{-1} \mathbf{A}_j + \mathbf{Q}_j^{-1}\right) + \sum_{i=1}^{K} \mathbb{E}[\alpha_{t+1,i}] \mathbf{C}_i^\top \mathbf{Q}_i^{-1} \mathbf{C}_i\right]^{-1} \tag{3.70}$$

$$\begin{aligned}\boldsymbol{\eta}_t =& \mathbf{V}_t \Bigg[\sum_{j=1}^{K} \mathbb{E}[\alpha_{t,j}] \left(\mathbf{A}_j^\top \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{y}_t - \boldsymbol{b}_t) + \mathbf{Q}_j^{-1} \mathbf{C}_j \mathbb{E}[\boldsymbol{x}_{t-1}]\right) \\ &+ \sum_{i=1}^{K} \mathbb{E}[\alpha_{t+1,i}] \mathbf{C}_i^\top \mathbf{Q}_i^{-1} \mathbb{E}[\boldsymbol{x}_{t+1}]\Bigg].\end{aligned} \tag{3.71}$$

3.4.3 M STEP

The parameters $\boldsymbol{\phi}$ are updated by maximizing the expected complete data loglikelihood with respect to the posterior distribution of $\boldsymbol{x}_t$ computed in the E-Step. This is achieved by solving the following formula:

$$\frac{\partial \mathbb{E}_{p(\boldsymbol{x}_{1:T}, Z_{1:T} | \boldsymbol{y}_{1:T})} [\mathcal{L}]}{\partial \phi} = 0, \tag{3.72}$$

where ϕ represents a parameter of $\boldsymbol{\phi}$ and $\mathcal{L}$ the complete data loglikelihood:

$$\begin{aligned}\mathcal{L} &= \log p\left(\boldsymbol{x}_{1:T}, Z_{1:T}, \boldsymbol{y}_{1:T}; \theta\right) \\ &\approx \sum_{t=1}^{T} \log \mathcal{N}(\boldsymbol{y}_t; \mathbf{A}_{Z_t} \boldsymbol{x}_t + \boldsymbol{b}_{Z_t}, \boldsymbol{\Sigma}_{Z_t}) + \sum_{t=2}^{T} \log \mathcal{N}(\boldsymbol{x}_t; \mathbf{C}_{Z_t} \boldsymbol{x}_{t-1}, \mathbf{Q}_{Z_t}) \\ &\quad + \sum_{t=2}^{T} \log \tau_{Z_{t-1}, Z_t} + \log p\left(\boldsymbol{x}_1, Z_1\right).\end{aligned} \tag{3.73}$$

The posterior distribution of $\boldsymbol{x}_t$ and Z_t being different for each approximation, the numerator in Equation (3.72) expression varies between the GPB2 and variational approximation, when one is using the GPB2 approximation the expected loglikelihood is defined as:

$$\mathbb{E}_{p(\boldsymbol{x}_{1:T}, Z_{1:T} | \boldsymbol{y}_{1:T})} [\mathcal{L}] \approx \mathbb{E}_{\prod_{t=1}^{T} p(Z_t | \boldsymbol{y}_{1:T})} \left[\mathbb{E}_{\prod_{t=1}^{T} p(\boldsymbol{x}_t | Z_t, \boldsymbol{y}_{1:T})} [\mathcal{L}]\right]. \tag{3.74}$$

and for the variational approximation:

$$\mathbb{E}_{p(\boldsymbol{x}_{1:T}, Z_{1:T} | \boldsymbol{y}_{1:T})} [\mathcal{L}] \approx \mathbb{E}_{\prod_{t=1}^{T} q^*(Z_t)} \left[\mathbb{E}_{\prod_{t=1}^{T} q^*(\boldsymbol{x}_t)} [\mathcal{L}]\right]. \tag{3.75}$$

§ Estimation of $\mathbf{C}_j$ and $\mathbf{Q}_j$

The update formulas of $\mathbf{C}_j$ and $\mathbf{Q}_j$ are obtained by using Equation (3.72) and replacing ϕ by $\mathbf{C}_j$ and $\mathbf{Q}_j$ and the expected loglikelihood by its value. For the approximation using

the GPB2 approach we use the expected loglikelihood defined in Equation (3.74) and we obtain the following expressions:

$$\begin{aligned}\mathbf{C}_j =& \left(\sum_{t=2}^{T} p(Z_t = j|\boldsymbol{y}_{1:T})\mathbb{E}[\boldsymbol{x}_t\boldsymbol{x}_{t-1}^\top]\right) \\ &\times \left(\sum_{t=2}^{T} p(Z_t = j|\boldsymbol{y}_{1:T})\mathbb{E}[\boldsymbol{x}_{t-1}\boldsymbol{x}_{t-1}^\top]\right)^{-1}, \qquad (3.76)\\ \mathbf{Q}_j =& \frac{1}{\sum_{t=2}^{T} p(Z_t = j|\boldsymbol{y}_{1:T})} \\ &\times \left(\sum_{t=2}^{T} p(Z_t = j|\boldsymbol{y}_{1:T}) \left(\mathbb{E}[\boldsymbol{x}_t\boldsymbol{x}_t^\top] - \mathbf{C}_j\mathbb{E}[\boldsymbol{x}_t\boldsymbol{x}_{t-1}^\top]\right)\right). \qquad (3.77)\end{aligned}$$

where:

$$\mathbb{E}\left[\boldsymbol{x}_t\boldsymbol{x}_{t-1}^T\right] = \mathbf{V}_{t,t-1} + \boldsymbol{\eta}_t\boldsymbol{\eta}_{t-1}^\top, \qquad (3.78)$$
$$\mathbb{E}\left[\boldsymbol{x}_{t-1}\boldsymbol{x}_{t-1}^T\right] = \mathbf{V}_{t-1} + \boldsymbol{\eta}_{t-1}\boldsymbol{\eta}_{t-1}^\top, \qquad (3.79)$$
$$\mathbb{E}\left[\boldsymbol{x}_t\boldsymbol{x}_t^T\right] = \mathbf{V}_t + \boldsymbol{\eta}_t\boldsymbol{\eta}_t^\top \qquad (3.80)$$

For our proposed variational approximation, we use the expected loglikelihood defined in Equation (3.75) and we obtain the following update formulas:

$$\begin{aligned}\mathbf{C}_j =& \left(\sum_{t=2}^{T} q^*\left(Z_t = j\right)\boldsymbol{\eta}_t\boldsymbol{\eta}_{t-1}^\top\right) \\ &\times \left(\sum_{t=2}^{T} q^*\left(Z_t = j\right)\mathbb{E}\left[\boldsymbol{x}_{t-1}\boldsymbol{x}_{t-1}^T\right]\right)^{-1}, \qquad (3.81)\\ \mathbf{Q}_j =& \frac{1}{\sum_{t=2}^{T} q^*\left(Z_t = j\right)} \\ &\times \left(\sum_{t=2}^{T} q^*\left(Z_t = j\right)\left(\mathbb{E}\left[\boldsymbol{x}_t\boldsymbol{x}_t^T\right] - \mathbf{C}_j\boldsymbol{\eta}_t\boldsymbol{\eta}_{t-1}^\top\right)\right). \qquad (3.82)\end{aligned}$$

§ Estimation of τ_{ij}

The estimation of the transition parameter τ_{ij} follows the same logic as the one for $\mathbf{C}_j$ and $\mathbf{Q}_j$ but adds the stochasticity constrain:

$$\begin{aligned}\tau_{ij} = \underset{\tau_{ij}}{\operatorname{argmax}}\, &\mathbb{E}_{p(\boldsymbol{x}_t|\boldsymbol{y}_{1:T})}\left[\mathcal{L}\right], \\ &\text{s.t.} \sum_{j=1}^{k} \tau_{ij} = 1. \qquad (3.83)\end{aligned}$$

To solve this we employ the Lagrange multiplier method. Finally by solving Equation (3.72) for τ_{ij} with the Lagrange multiplier we obtain the following expression for the update of τ_{ij}, with the GPB2 approximation:

$$\tau_{ij} = \frac{\sum_{t=2}^{T} p(Z_t = j|\boldsymbol{y}_{1:T}) p(Z_{t-1} = i|\boldsymbol{y}_{1T})}{\sum_{t=2}^{T} p(Z_{t-1} = i|\boldsymbol{y}_{1:T})} \tag{3.84}$$

and with our variational approximation:

$$\tau_{ij} = \frac{\sum_{t=2}^{T} q^*(Z_t = j) q^*(Z_{t-1} = i)}{\sum_{t=2}^{T} q^*(Z_{t-1} = i)} \tag{3.85}$$

The complete derivation for the E-step and M-step for both approximation (GPB2 and variational) can be found in Appendix B.

§ Discussion

Even though the two EM procedures follow a similar procedure, the resulting formulas to update the parameters $\boldsymbol{\phi}$ are different. In the formulas to update $\mathbf{C}_j$ (respectively $\mathbf{Q}_j$), with our proposed variational approximation $\mathbb{E}\left[\boldsymbol{x}_t \boldsymbol{x}_{t-1}^\top\right]$ in Equations (3.76) and (3.77), is replaced by $\boldsymbol{\eta}_t \boldsymbol{\eta}_{t-1}^\top$ in Equations (3.81) and (3.82), respectively. This is due to the design of our variational approximation, by breaking the temporal depency, we remove dependencies between variables. While this might affect the accuracy it also reduces the computation complexity by avoiding to compute the joint covariance of $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t-1}$.

3.5 EXPERIMENTAL VALIDATION

In this section we evaluate the performance of the tracking methods presented in Section 3.2. Experiments are carried out on two publicly available datasets, the Biwi Kinect head pose dataset [Fanelli 13] and the Vernissage dataset [Jayagopi 12]. First We conduct experiments to compare the performances of the two approaches against each other and also against non tracking method of Chapter 2. In a second time we compare against other tracking methods to evaluate the advantages of the model with respect to classic tracking technics.

3.5.1 FACE REPRESENTATION

To gauge the performance of the proposed method we used two datasets: the Biwi Kinect head pose dataset [Fanelli 13], previously described with details in Chapter 2 Section 2.4.4 and the Vernissage dataset. The Vernissage dataset, [Jayagopi 12], consists of ten recordings of people in an exhibition. Each recording is composed of two people. The dataset is composed of ten-minute recordings involving 20 different persons. The scene was recorded with a camera mounted onto the robot head and with a network of infrared cameras placed on the walls. These cameras are used in conjunction with optical markers, placed onto both the robot and person heads, to provide accurate head positions and head orientations in a common reference frame. The robot-head camera is synchronized with the infrared cameras at 25 FPS, hence there is a total of 90, 000 frames.

Face regions are extracted from images with a face detector [Viola 01]. This detector is efficient and robust with both frontal- and side-views of faces. Using the detection we run a face tracker using particle filtering to extract a face at each frame of the videos. Nevertheless, the obtained face regions are noisy, i.e. the bounding boxes are not always nicely aligned onto the faces. This yields extremely realistic input data for the tested methods.

From each face region thus detected, we extract a feature vector. We conducted the experiments using HoG based features. The HoG based features are obtained by computing with several different cell resolutions, namely 32×32, 16×16 and 8×8 pixels, with block size of 2×2 cells and 8 bins to quantize the gradient orientation. This results in a pyramidal model that is represented by a feature vector of size $D = 1888$.

3.5.2 PROTOCOL

The regression parameters $\boldsymbol{\theta}$ (Equation (2.6)) and the filtering parameters $\boldsymbol{\phi}$ (Equation (3.7)) are learned separately. First, the regression parameters θ are estimated using the EM algorithm described in Appendix B. Second, the filtering parameters $\boldsymbol{\phi}$ are estimated using the methods described in 3.2. For all experiments K is fixed to be equal to 25, as it provides lower error and less parameters to estimate compared to other values of K, Figures 2.4 and 2.5. The parameters $\boldsymbol{\theta}$ are intitialized using GMM model. The parameters

Table 3.1: Comparison between the GPB2 and Variational approximation, for each angle and complexity (Cpx), average time in second use for 1 time step implementation on Matlab using Intel Xeon CPU

	Pitch		Yaw		Roll		Time
	Avg.	Std.	Avg.	Std.	Avg.	Std.	
GLLiM	10.54	13.38	11.15	17.93	5.23	5.99	–
GPB2	**9.03**	**10.89**	**8.77**	**13.42**	***4.75***	***5.11***	9.55
Variational	***9.25***	***11.21***	***9.10***	***14.9***	**4.44**	**4.76**	**3.45**

$\left\{\mathbf{C}_j, \mathbf{Q}_j\right\}_{j=1}^{K}$ of $\boldsymbol{\phi}$ are initialized with identity matrices and the transition matrix $\{\tau_{ij}\}_{i,j=1}^{K}$ is initialized with the Bhattacharrya distance [Bhattacharyya 43] between two subspaces obtained using the parameters $\boldsymbol{\theta}$ of the low dimensional space defined by Z. For the variational approximation, $q(\boldsymbol{x}_t)$ is needed to compute $q(Z_t)$ thus we initialize the parameters $\boldsymbol{\eta}_t$ and $\mathbf{V}_t$ of $q(\boldsymbol{x}_t)$ as follows:

$$\boldsymbol{\eta}_t = \boldsymbol{\eta}_{t-1}, \mathbf{V}_t = \mathbf{V}_{t-1}, \tag{3.86}$$

after computing $q(Z_t)$ we update $q(\boldsymbol{x}_t)$. The performance are measured using the absolute error to the ground truth, mean (MAE) and standard deviation (Std.) to compare the estimation between methods.

3.5.3 Evaluation of the two approaches

Table 3.1 summarizes the comparison of the GBP2 and variational approximations in terms of accuracy and complexity on the Biwi Kinect dataset. Both tracking methods reduce the average estimation error and standard deviation with respect to the static estimation. Figure 3.3 shows the estimation for yaw angle on a full sequence of each tracking approximation compared to the estimation with GLLiM. Over a sequence the tracking methods make the estimation closer to the ground truth and smoother compared to the estimation with GLLiM, this can be seen in Figure 3.4 as well. The GBP2 approximation gives more accurate estimation than the variational approximation, but the time complexity is much higher than the variational approximation. The estimation of the filtering distribution is more than two times faster with the variational approximation than the GPB2. For the problem of head-pose estimation with $K = 25$, the accuracy gained by using approximation with the GPB2 algorithm is not so much compared to the one gained with the variational approximation. But the computational time saved with the variational approximation is non negligeable. In a case where K is much bigger using the variational approximation over the GBP2 one will be more relevant. The results of Table 3.1 confirm what was described in [Pavlovic 00].

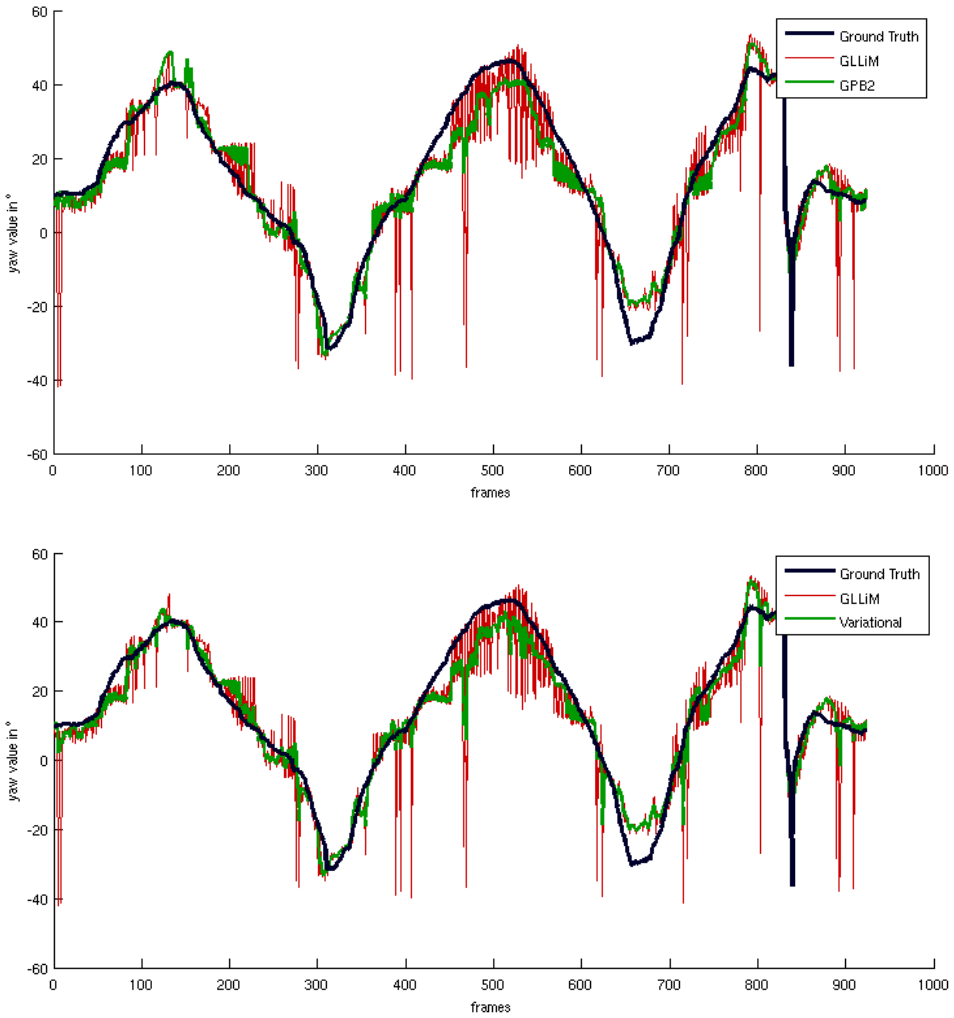

Figure 3.3: Comparison between the estimated yaw angle (top) and yaw angle (bottom) with three different methods: HPE_GLLiM (red), Kalman Filter (blue), and the proposed HPE_SKF (green) for the Biwi-Kinect dataset.

Table 3.2: Average (Avg.) and standard deviation (Std.) of the absolute error (in degrees) for the pitch, yaw and roll angles (when applicable) on the Biwi Kinect dataset. Head bounding boxes are extracted using a face detection algorithm.

	Pitch		Yaw		Roll	
Methods	Avg.	Std.	Avg.	Std.	Avg.	Std.
[Uřičář 12]	13.12	10.79	21.1	14.16	–	–
openFace [Baltrušaitis 16]	***9.23***	15.69	29.43	25.74	10.72	11.33
[Drouard 17b]	10.54	13.38	11.15	17.93	5.23	5.99
[Drouard 17b] + KF	10.35	13.19	10.97	17.75	5.12	5.93
SKF_GPB2	**9.03**	**10.89**	**8.77**	**13.42**	***4.75***	***5.11***
SKF_variational	***9.25***	***11.21***	***9.10***	***14.9***	**4.44**	**4.76**

3.5.4 BENCHMARK

The proposed model is compared to the following tracking methods: (i) a landmark-based approach that uses the facial landmark localization method of [Uřičář 12] (Flandmarks) combined with 2D-to-3D landmark-based pose estimation method, namely the PnP (perspective n-point) algorithm available with OpenCV, (ii) a second landmark-based approach [Baltrušaitis 16] (iii) the GLLiM-based method presented in Chapter 2 and in [Drouard 15] which is referred to as HPE-GLLiM, and (iv) the regression method [Drouard 15] combined with a standard Kalman filter [Arulampalam 02, Bishop 07].

Table 3.3: Average (Avg.) and standard deviation (Std.) of the absolute error (in degrees) for the pitch and yaw angles on the Vernissage dataset. Head bounding boxes are extracted using a face detection algorithm combined with a face tracker.

	Pitch		Yaw	
Methods	Avg.	Std.	Avg.	Std.
openFace [Baltrušaitis 16]	21.3	24.82	13.18	10.67
[Drouard 17b]	22.94	21.49	12.28	9.42
[Drouard 17b] + KF	22.92	21.49	12.27	9.41
SKF_GPB2	**19.96**	**19.28**	***11.71***	***8.78***
SKF_variational	***20.04***	***19.97***	**10.77**	**7.86**

The results (average and standard deviations of the absolute error) obtained are shown

in Table 3.2. The proposed tracking method improves head-pose parameter tracking, compared to all the other methods. For example, the average error for the yaw angles in Table 3.2 is of $8.77°$ using GPB2 approximation and $9.10°$ using variational approximation while all other methods yield an error larger than $10°$. We observe the same behavior for the pitch and roll angles. Moreover, our method also reduces the standard deviation. In Table 3.3 we also observed that our tracking approaches reduces the estimation error and standard deviation on the Vernissage dataset, though on the Vernissage dataset both approaches yield really close results. Estimation using GPB2 approach gives a slightly better estimation for the pitch angle while our approach using variational approximation gives a better estimation for the yaw angle. Compared to the landmark-based method of [Uřičář 12], the proposed method is able to provide an estimation for each test input, whereas the method based on landmarks is unable to provide an output when some of the landmarks are not visible due to extreme head orientations. In this case [Uřičář 12] yields very large errors, e.g. first row of Table 3.2. We also note that the proposed HPE_SKF method performs much better than a standard Kalman filter. From the results presented in Tables 3.1, 3.2 and 3.3 for the problem of head-pose tracking, using the variational approximation seems the optimal solution since it provides a estimation really close to the one using GBP2 approximation with a smaller time complexity. The results presented are preliminary, and we wish to extend them by testing on more challenging datasets and also to use deep features obtained using convolutional neural networks.

3.6 Conclusion

In this chapter, we extented the model presented in Chapter 2 in order to incorporate temporal information. We combined the mixture of affine transformations with a dynamic model, the latter being also a mixture model. This temporal model is defined as a switching Kalman filter, it can be seen as a mixture of Kalman filters with the possibility to switch between filters at each time step. This possibility of switching makes the model intractable, the number of components increases exponentially with time. We derived two approaches to overcome this issue. The first one using the GPB2 algorithm that combines components of the mixture together. The second one being a variational approximation that forces the model to keep a fixed number of components at each time step, by breaking the time dependencies in the model.

Trackers have the advantage of combining information from both past and present, and hence they avoid oscillations between consecutive estimations simply based on independent observations. Overall, the output of the tracking methods (SKF with GPB2 and variational) presented are both more accurate and smoother than the output of several head-pose methods (with and without tracking). Moreover, noisy observations, e.g. due to badly aligned bounding boxes or to partial occlusions, do not impact too much the proposed trackers because the temporal models, once properly trained, do not allow oscillations between consecutive estimations. In the future, we wish to use CNN based features for the method. They have proved to be really efficient for computer vision problem, as detection, recognition and pose estimation. They might give a better representation of the

faces and improve the accuracy of the model and the robustness to the estimation.

One publication emerges from this work [Drouard 17a] in the IEEE Winter Conference on Applications of Computer Vision in 2017.

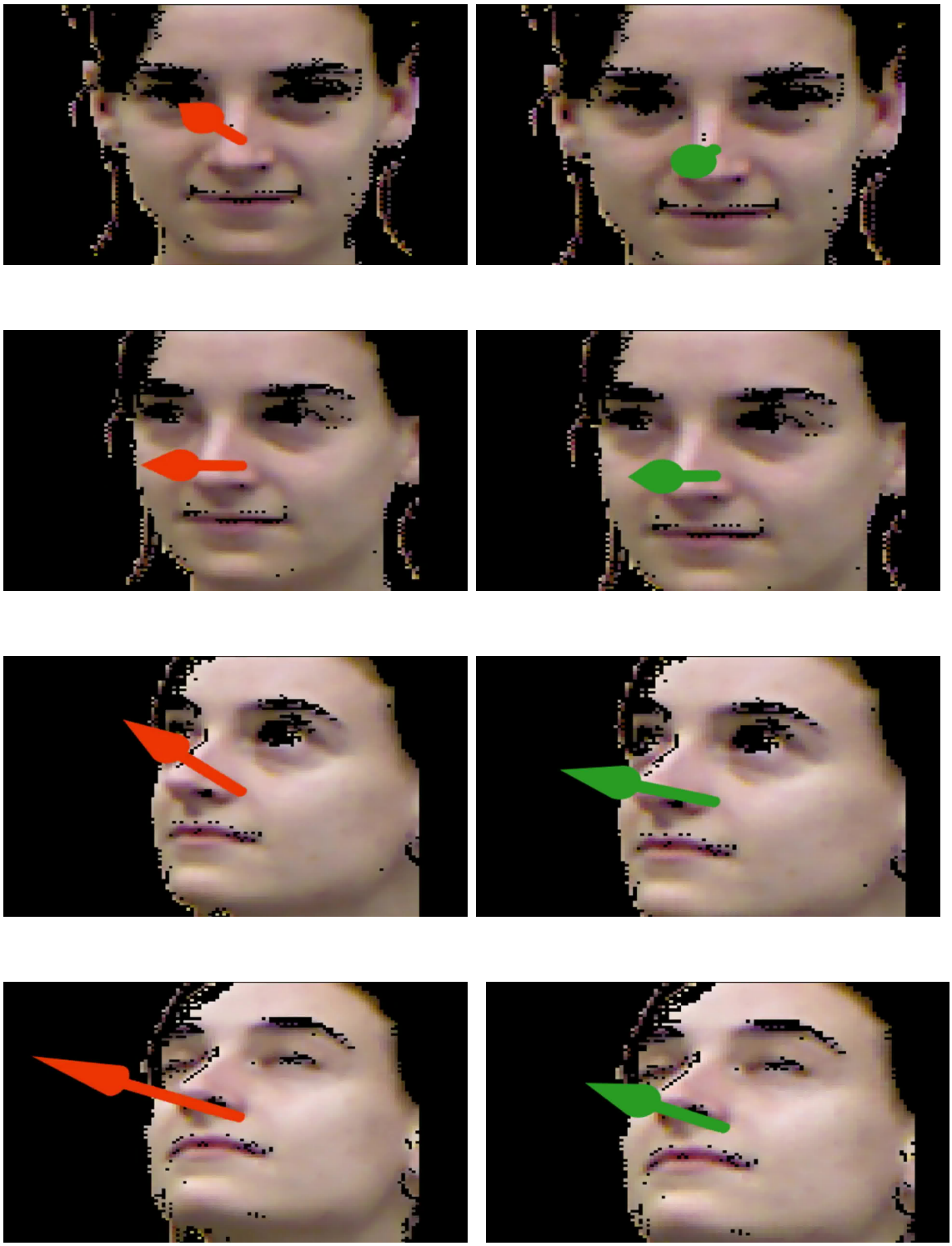

Figure 3.4: Results of the tracking method (left column) and the estimation method of Chapter 2 (right column) on a sequence of the Biwi dataset

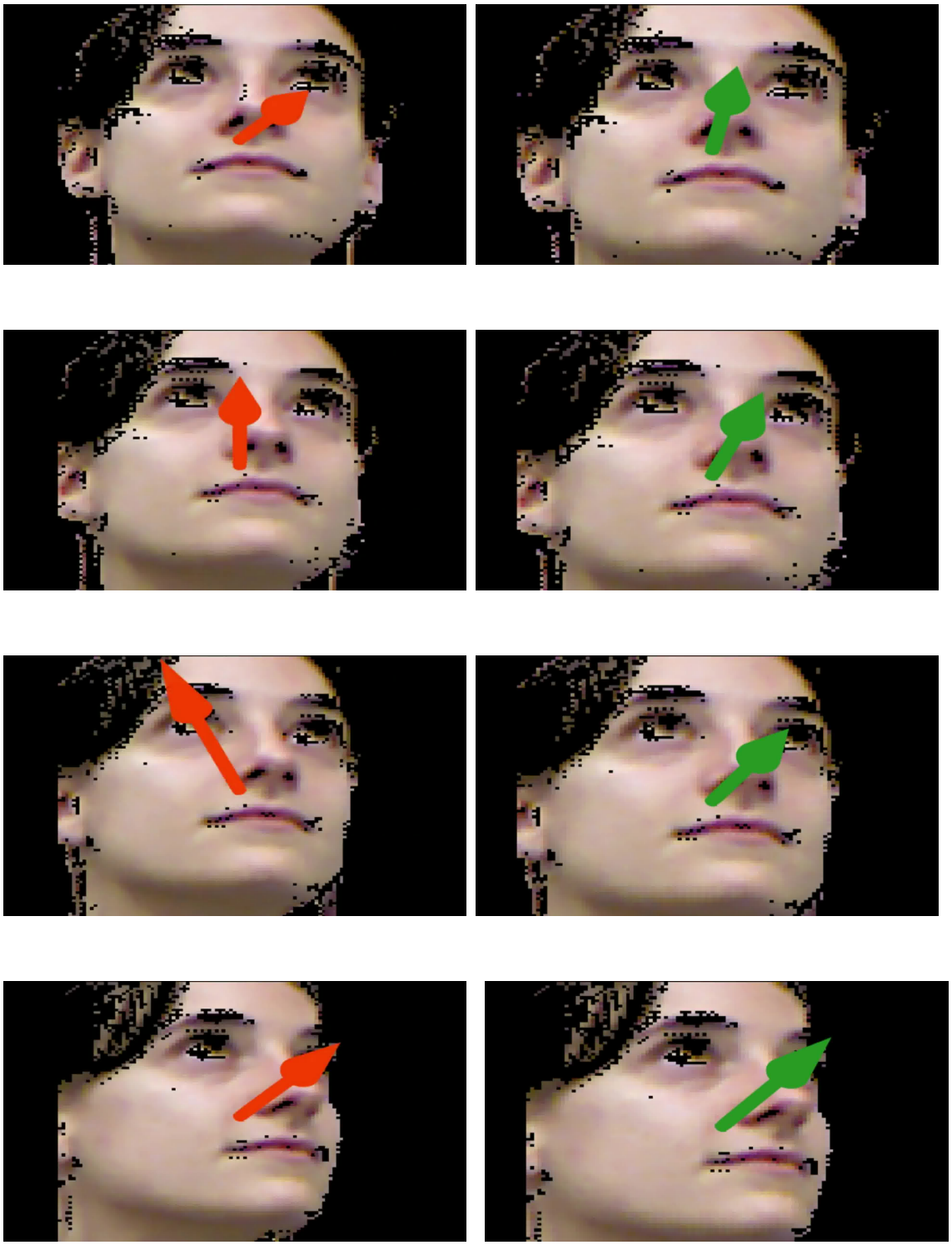

Figure 3.4: Results of the tracking method (left column) and the estimation method of Chapter 2 (right column) on a sequence of the Biwi dataset

CHAPTER 4

TEMPORAL MODEL FOR SPEAKER TRACKING

Originally the GLLiM model was developped for infering the direction of arrival (DOA) of a sound source in an image using binaural features from a pair of microphones, [Deleforge 14]. With respect to the *generic* version of the GLLiM model presented in Chapter 2, we modify the observation model to compensate for the sparsity of the audio observations. We formally derive the equations of the switching linear regression model to integrate the new observation model adapted to the problem of DOA estimation. The learning procedure to estimate the model parameters reuses the one for the generic version presented in Chapters 2 and 3.

4.1 NATURAL SPEECH REPRESENTATION

In audio processing working directly with signals recorded by microphones is not always the most efficient way. Most of the applications use a time-frequency representation since it conveys richer information than time representation and also is independent from the speech content. The time-frequency representation is obtained using *short term Fourier representation* (STFT). In this study we are using a pair of microphones, namely left and right microphone. So for each microphone we compute the associated complex-value spectrogram on audio signal of 320ms, we used a window length of 128ms with an overlap of 87.5% to compute the STFTs. The second step consists in computing the interaural level and phase differences (ILD and IPD) between the left and right microphone spectrograms. The matrix containing the phase difference values is transformed into 2 matrices one for the real part of the phase and one for the imaginary part. This change is made to not work with circular data. The resulting will be two real values matrices that will be concatenated together, thus forming a single matrix of size $(D \times S)$ where D represents the number of frequency bins and is equal to 1534, and S the number of time-window

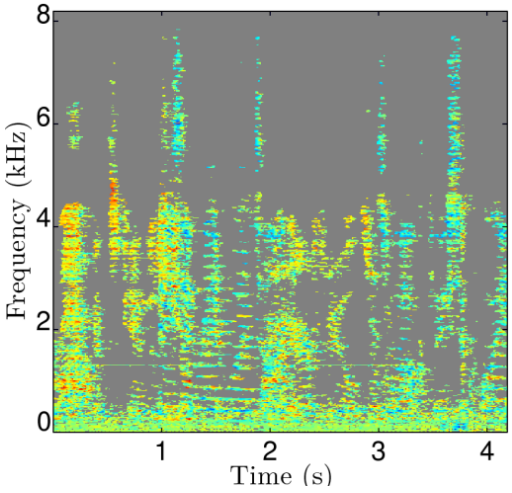

Figure 4.1: A recorded natural speech spectrogram, gray area means frequency is inactive at the time frequency point

used to compute the STFTs. When the audio signal is a recording of a natural speech most of the frequencies will be inactive (i.e. the energy of the signal at these frequencies will be null), see Figure 4.1.

The inconvenient now resides in the physical property of a natural speech signal. For speech signals, time-frequency representations are extremely sparsed, i.e. many time-frequency points are null and are unusable (considered as missing values). To only work with active time-frequency points of the speech signal we build a binary mask by thresholding the energy of the resulting STFT matrix. The binary mask has the same size as STFT matrix and each of its entries is equal to 0 if the value is null and 1 otherwise.

4.2 NEW OBSERVATION MODEL

Sound signals are described by a time series $\mathbf{Y}$ of length S, namely $\mathbf{Y} = \{\boldsymbol{y}_1, \dots \boldsymbol{y}_S\}$ with $\boldsymbol{y}$ a vector of dimension $D = 1534$. This time series $\mathbf{Y}$ is viewed as a $D \times S$ matrix where each entry point $\boldsymbol{y}^{d,s}$ represents a time-frequency point. The associated binary mask, that indicates active time-frequency points, is called χ and has the same size as $\mathbf{Y}$. The audio feature is now represented by $\{\mathbf{Y}, \chi\} = \{y_{d,s}, \chi_{d,s}\}_{d,s=1}^{D,S}$. The observation model presented in Chapter 2 can not be used as it is anymore. If we assume that each time-frequency points of the matrix are independent to each other, the observation distribution

$p(\boldsymbol{y}_t|\boldsymbol{x}_t, Z_t = j) = \mathcal{N}(\boldsymbol{y}_t; \mathbf{A}_j\boldsymbol{x}_t + \boldsymbol{b}_j, \boldsymbol{\Sigma}_j)$ can be replaced by:

$$p(\{\mathbf{Y}_t, \chi_t\}|\boldsymbol{x}_t, Z_t = j) = \prod_{d,s}^{D,S} p(y_t^{d,s}|\boldsymbol{x}_t, Z_t = j)^{\chi_t^{d,s}}, \tag{4.1}$$

$$= \prod_{d,s}^{D,S} \mathcal{N}\left(\boldsymbol{y}_t^{d,s}|; \boldsymbol{a}_{d,j}\boldsymbol{x}_t + b_{d,j}, \sigma_{d,j}\right)^{\chi_t^{d,s}}. \tag{4.2}$$

4.3 INTEGRATING THE NEW OBSERVATION MODEL

We are now trying to express the tracking model presented in Chapter 3 Equation 3.20 for the new observation model. The derivations to obtain the filtering forward distribution are similar to the ones presented in Chapter 3, starting from Equation (3.10):

$$p(\boldsymbol{x}_t|\{\mathbf{Y}, \chi\}_{1:t}) = \sum_{j=1}^{K}\sum_{i=1}^{K} \tau_{ij}\nu_{t-1}^i \mathcal{N}(\boldsymbol{y}_t|\mathbf{A}_j\boldsymbol{x}_t + \boldsymbol{b}_j, \boldsymbol{\Sigma}_j) \int_{\boldsymbol{X}_{t-1}} \mathcal{N}(\boldsymbol{x}_t|\mathbf{C}_j\boldsymbol{x}_{t-1}, \mathbf{Q}_j)\mathcal{N}(\boldsymbol{x}_{t-1}|\boldsymbol{\nu}_{t-1}^i, \mathbf{V}_i)d\boldsymbol{x}_{t-1}, \tag{4.3}$$

we replace the observation distribution by the new one defined in Equation (4.2):

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \sum_{j=1}^{K}\sum_{i=1}^{K} \tau_{ij}\nu_{t-1}^i \prod_{d,s}^{D,S} \mathcal{N}\left(\boldsymbol{y}_t^{d,s}|\boldsymbol{a}_{d,j}\boldsymbol{x}_t + b_{d,j}, \sigma_{d,j}\right)^{\chi_t^{d,s}} \times \int_{\boldsymbol{X}_{t-1}} \mathcal{N}(\boldsymbol{x}_t|\mathbf{C}_j\boldsymbol{x}_{t-1}, \mathbf{Q}_j)\mathcal{N}(\boldsymbol{x}_{t-1}|\boldsymbol{\nu}_{t-1}^i, \mathbf{V}_i)d\boldsymbol{x}_{t-1} \tag{4.4}$$

$$= \sum_{j=1}^{K}\sum_{i=1}^{K} \tau_{ij}\nu_{t-1}^i \prod_{d,s}^{D,S} \mathcal{N}\left(\boldsymbol{y}_t^{d,s}|\boldsymbol{a}_{d,j}\boldsymbol{x}_t + b_{d,j}, \sigma_{d,j}\right)^{\chi_t^{d,s}} \times \mathcal{N}(\boldsymbol{x}_t|\mathbf{C}_j\boldsymbol{\eta}_{t-1}^i, \mathbf{Q}_j + \mathbf{C}_j\mathbf{V}_i\mathbf{C}_j^\top). \tag{4.5}$$

The exponential part of the product of Gaussian distributions is rearranged as follows:

$$\sum_{d,s=1}^{D,S} \frac{\chi_t^{d,s}}{\sigma_{d,j}^2}\left(y_t^{d,s} - \boldsymbol{a}_{d,j}\boldsymbol{x}_t - b_{d,j}\right)^2 + \left(\boldsymbol{x}_t - \mathbf{C}_j\boldsymbol{\eta}_{t-1}^i\right)^\top \left(\mathbf{Q}_j + \mathbf{C}_j\mathbf{V}_i\mathbf{C}_j^\top\right)^{-1}\left(\boldsymbol{x}_t - \mathbf{C}_j\boldsymbol{\eta}_{t-1}^i\right)$$

$$= \boldsymbol{x}_t^\top\left(\sum_{d,s=1}^{D,S} \frac{\chi_t^{d,s}}{\sigma_{d,j}^2}\boldsymbol{a}_{d,j}^\top\boldsymbol{a}_{d,j} + \left(\mathbf{Q}_j + \mathbf{C}_j\mathbf{V}_i\mathbf{C}_j^\top\right)^{-1}\right)\boldsymbol{x}_t - 2\boldsymbol{x}_t^\top\left(\frac{\chi_t^{d,s}}{\sigma_{d,j}^2}\boldsymbol{a}_{d,j}^\top\left(y_t^{d,s} - b_{d,j}\right)\right) + c, \tag{4.6}$$

with c being a residual that doesn't contain $\boldsymbol{x}_t$. Following this, the product of Gaussian distributions can be rewritten as a single Gaussian distribution over $\boldsymbol{x}$ times a term C and

is expressed as:

$$\prod_{d,s}^{D,S} \mathcal{N}\left(\boldsymbol{y}_t^{d,s} | \boldsymbol{a}_{d,j}\boldsymbol{x}_t + b_{d,j}, \sigma_{d,j}\right)^{\chi_t^{d,s}} \mathcal{N}(\boldsymbol{x}_t | \mathbf{C}_j\boldsymbol{\eta}_{t-1}^i, \mathbf{Q}_j + \mathbf{C}_j\mathbf{V}_i\mathbf{C}_j^\top) = C \times \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_{t|t-1}^{ij}, \mathbf{W}_{t|t-1}^{ij}), \tag{4.7}$$

with the parameters of the Gaussian distribution equals to:

$$\mathbf{W}_{t|t-1}^{ij} = \left(\sum_{d,s=1}^{D,S} \frac{\chi_t^{d,s}}{\sigma_{d,j}^2}\boldsymbol{a}_{d,j}^\top\boldsymbol{a}_{d,j} + \left(\mathbf{Q}_j + \mathbf{C}_j\mathbf{V}_i\mathbf{C}_j^\top\right)^{-1}\right)^{-1} \tag{4.8}$$

$$\boldsymbol{\mu}_{t|t-1}^{ij} = \mathbf{W}_{t|t-1}^{ij}\left(\frac{\chi_t^{d,s}}{\sigma_{d,j}^2}\boldsymbol{a}_{d,j}^\top\left(y_t^{d,s} - b_{d,j}\right) + \left(\mathbf{Q}_j + \mathbf{C}_j\mathbf{V}_i\mathbf{C}_j^\top\right)^{-1}\mathbf{C}_j\boldsymbol{\eta}_{t-1}^i\right) \tag{4.9}$$

and the term C:

$$C = \prod_{d,s=1}^{D,S} \frac{1}{\sqrt{2\pi\sigma_{d,j}^2}} \frac{\sqrt{2\pi|\mathbf{W}_{t|t-1}^{ij}|}}{\sqrt{2\pi|\mathbf{Q}_j + \mathbf{C}_j\mathbf{V}_{t-1}^i\mathbf{C}_j^\top|}} \times \exp\left(-\frac{1}{2}\left(\sum_{d,s=1}^{D,S}\frac{\chi_{d,s}}{\sigma_{d,j}^2}\left(y_{d,s} - b_{d,j}\right) + \boldsymbol{\eta}_{t-1}^i\mathbf{C}_j^\top\left(\mathbf{Q}_j + \mathbf{C}_j\mathbf{V}_i\mathbf{C}_j^\top\right)^{-1}\mathbf{C}_j\boldsymbol{\eta}_{t-1}^i - \boldsymbol{\mu}_{t|t-1}^{ij}\mathbf{W}_{t|t-1}^{ij}\boldsymbol{\mu}_{t|t-1}^{ij}\right)\right). \tag{4.10}$$

Thus the filtering distribution is now expressed as:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \sum_{j=1}^{K}\sum_{i=1}^{K}\pi_{t|t-1}^{ij}\mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_{t|t-1}^{ij}, \mathbf{W}_{t|t-1}^{ij}), \tag{4.11}$$

where:

$$\pi_{t|t-1}^{ij} = \tau_{ij}\nu_{t-1}^i \times C \tag{4.12}$$

Here also the filtering distribution number of components is multiplied by K over a time step. The GPB2 algorithm is applied to reduce the number of components in the mixture to K to finally obtain:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \sum_{j=1}^{K}\sum_{i=1}^{K}\rho_t^j\mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\eta}_t^j, \mathbf{V}_t^j), \tag{4.13}$$

where $\boldsymbol{\eta}_t^j$, $\mathbf{V}_t^j$ and ρ_t^j are obtained from $\boldsymbol{\mu}_{t|t-1}^{ij}$, $\mathbf{W}_{t|t-1}^{ij}$ and $\pi_{t|t-1}^{ij}$ using Equations (3.62) to (3.64).

Figure 4.2: popeye

4.4 EXPERIMENTAL VALIDATION

Experiments were run on the AVDIAR dataset, a dataset for audio-visual analysis of conversational scenes recorded by members of the Perception team at Inria Grenoble [Gebru 17]. The dataset comprises 5 scenarios recorded in two different environments, a living room and a meeting room, each scenario is recorded multiple times with a different set of participants. The recordings were done using the Popeye recording setting, a dummy head with 6 microphones and two wide-angle cameras attached to it. Figure 4.2 shows the disposition of cameras and microphones on the dummy head.

For the experiments we used one scenario of the dataset in which a single person is speaking and moving at the same time in the field of view of the camera. Six recordings of this scenario were made, 4 in the living room condition and 2 in the meeting room, subjects were equally distributed between men and women. The dataset provides annotations for the face positions, this was used to determine the ground truth position of the sound source. The model parameters $\boldsymbol{\theta}$ are learned using the EM-algorithm of Appendix A and white noise sounds, because they have the property to be activated at each frequency. The dataset provides 1600 white noise recordings and their associated ground truth positions

for each room. The positions span the all fields of view with depth of 2 and 3 meters away from the popeye robot. The value of K was set to 16. During the testing, each video were split into sequences were the person was speaking and for each sequence a sliding window of 320 ms (8 visual frames) was used to determine the sound DOA, the window is shifted of 40 ms (1 visual frame) for each time step. Binaural features were computed on this 320 ms signal window. Sound signals were sampled at a frequency of $16KHz$, from the 320 ms signal, spectrogram is computed with a sliding window of 128 ms and an overlap of 87.5%. The size of the final observations is 3073×21.

The performance is measured in terms of absolute pixel error to the face center, images were recorded in Full HD format, i.e. with a width of 1920 pixels, the field of view of the images span 97° approximately. Because we used only 1 pair of microphones that are positioned at the same height, thus only the azimuth of the sound direction of arrival is estimated. The method is compared to the estimation GLLiM regression, the GLLiM regression combined with a Kalman filter. Table 4.1 summarizes the results, with average and standard deviation of the absolute error. As in the head-pose tracking experiments the same behavior is observed, the tracking model helps to improve the estimation using GLLiM in terms of average error and standard deviation. The tracking method keeps information about previous position, thus being less affected by reverbations that are affecting estimations. The error is also lower compared to the Kalman filter.

Table 4.1: Average (Avg.) and standard deviation (Std.) of the absolute error (in pixel and degrees) for the azimuth of the direction of arrival of the sound.

	Azimuth			
	Pixels		degrees	
Methods	Avg.	Std.	Avg.	Std.
SSL_GLLiM [Deleforge 14]	107.2	135.6	6.89	8.06
Kalman filter [Arulampalam 02, Bishop 07]	***106.8***	***135.1***	***6.87***	***8.03***
SSL_SKF	**91.3**	**105.2**	**5.97**	**6.46**

Figure 4.3 shows visualization of the results on consecutive frames. Ground truth and estimations using GLLiM and the switching Kalman filter are displayed using a vertical line to represent the azimuth of the sound direction or arrival in the image. As in the case of head-pose estimation, the estimation with the tracking method stays closer to the ground truth and correct bad estimation due to noisy observation.

4.5 CONCLUSION

In this chapter, we demonstrate the ability of the switching dynamical model to be applied to various unrelated problems (head pose tracking and sound direction of arrival

tracking). Starting from our previous work on sound source direction of arrival estimation in [Deleforge 14], we adjust the observation model presented in Chapters 2 to take into account the sparsity of natural speech signals. This new observation model is combined with the tracking framework of Chapter 3 to provide a sound DOA tracking from sparse observations. The results we obtained on speaker DOA tracking follow the same tendency as for the head-pose tracking problem, the temporal model helps to smooth and improve the estimation over a temporal sequence of observations. Tracking speaker from audio can help to overcome the limitation of visual tracking. The audio *field of view* being larger than the visual *field of view* (except with a 360 camera), it provides information about the position of a speaker when he is outside the field of view. In the future we plan to use this for multi-person tracking with companion robot.

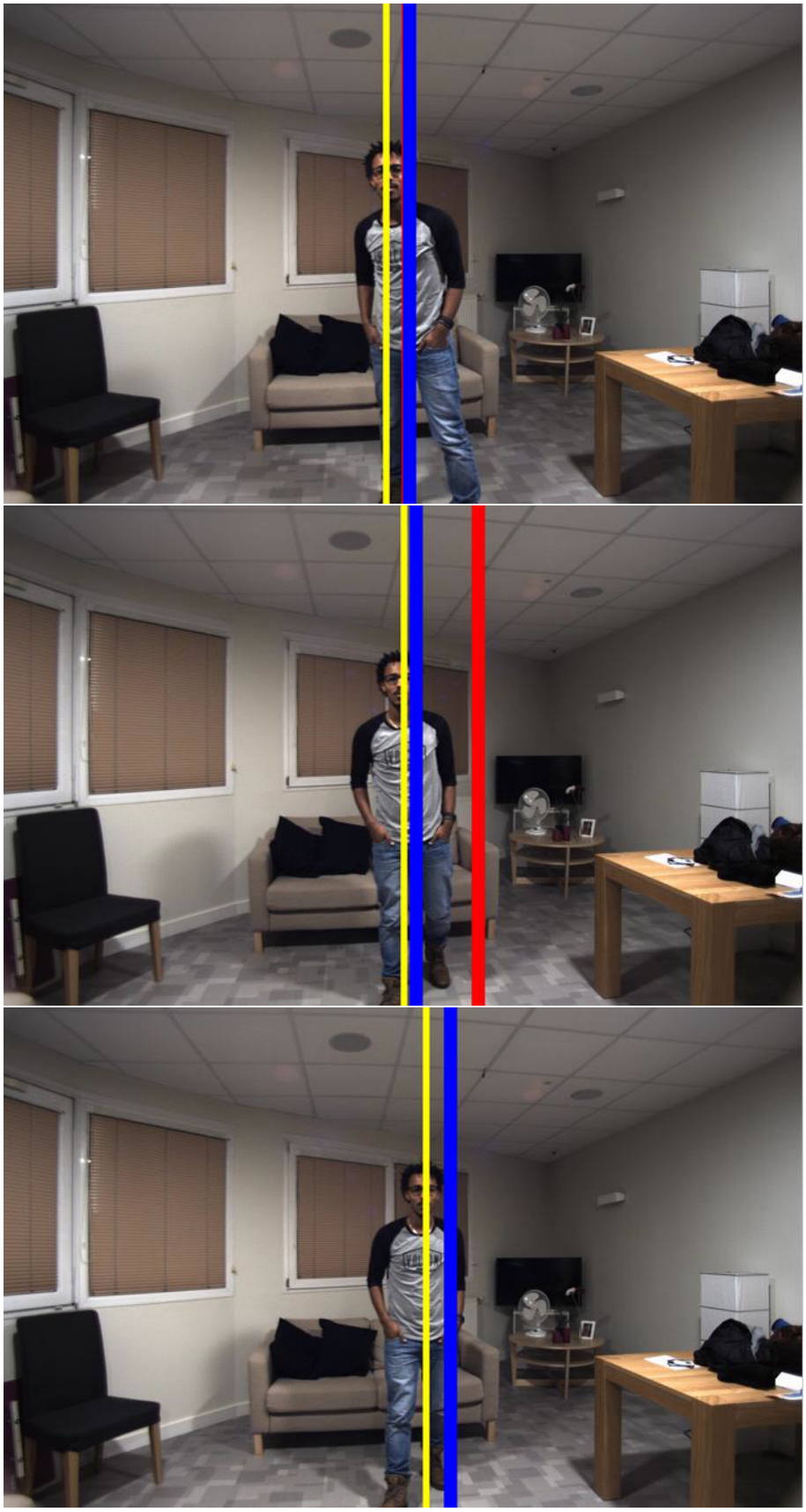

Figure 4.3: Results of the tracking method for DOA tracking, yellow line indicates ground truth position, red DOA estimation using method from [Deleforge 14], blue results using the tracking method

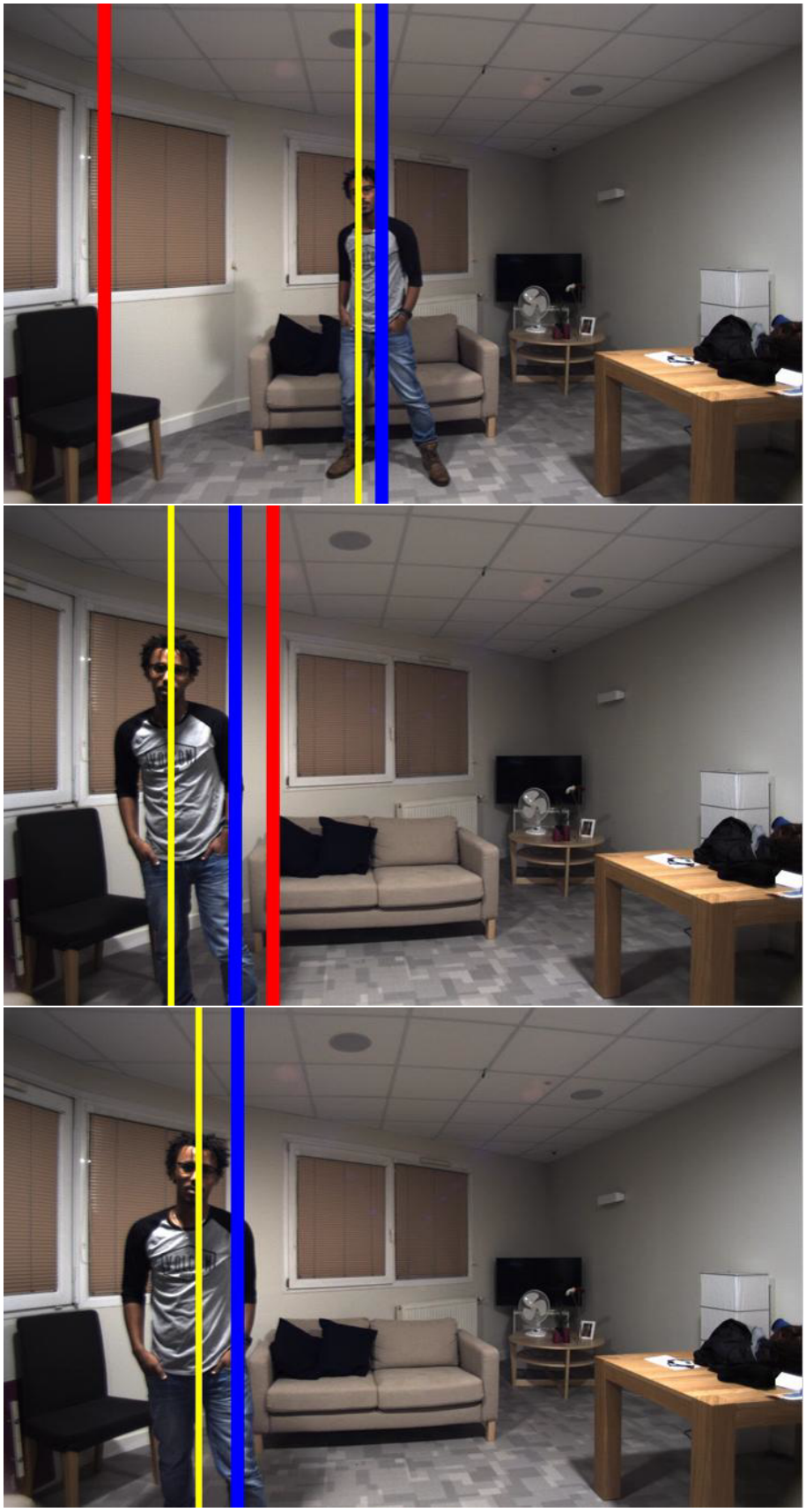

Figure 4.3: Results of the tracking method for DOA tracking, yellow line indicates ground truth position, red DOA estimation using method from [Deleforge 14], blue results using the tracking method

CHAPTER 5

APPLICATION TO ROBOTICS

In the context of the EARS project with the engineers of the Perception Team we developed a speaker localization module for the Nao robot. The module is using the NaoLab framework, [Badeig 15], to send and receive information from the robot. The speaker localization module is running in real-time and implemented with Matlab. We choose Matlab because of the demand of the partners for using methods implemented with Matlab for the robot.

5.1 THE NAO ROBOT

The Nao robot is a small humanoid companion robot, it has been developed for human-robot interaction tasks. For this work only the head was used, the rest of the body was static. Its head is equipped with 4 microphones, 2 cameras and 2 loud speakers, the position of all these elements on the head can be found in Figure 5.1.

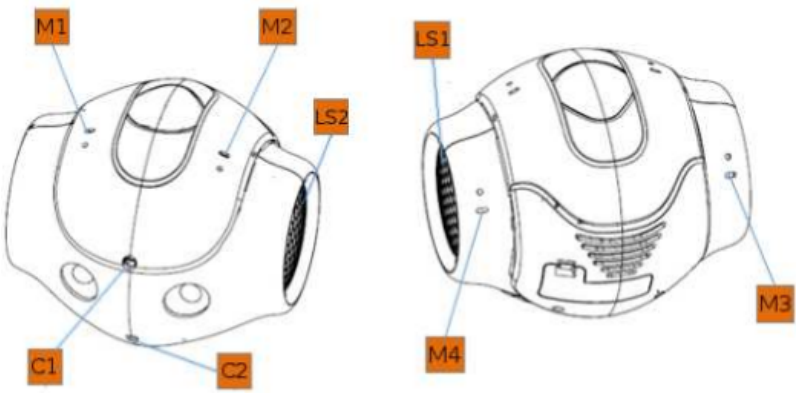

Figure 5.1: Configuration of the head of the Nao robot. M1-4 are the microphones positions, C1-2 the cameras ones and LS1-2 the loud speakers ones

5.2 FOLLOWING A SPEAKER WITH THE NAO ROBOT

Our goal was to allow Nao to find a speaker in a scene. We developed a module that combines both audio and visual cues to achieve this task. Combining these two cues will help to reduce the limitation of each one. Indeed the field of view of the robot is quite small, thus using audio cues helps to enlarge the visual field of view to find people using the sound they emits. And because the visual detection is more precise to find a person than just using the sound that the person emits, using audio and vision will result in a more robust speaker localization method. The method was running using Matlab on a remote computer. The computer was receiving audio and video feed from the Nao, processing them to estimate the position of the speaker and then sending commands to the robot's head in order to make him turn it head to face the speaker. The front microphones were used and because they are at the same heights only the azimuth of the direction of arrival of the sound can be estimated, elevation was estimated using face detector in the region of the sound direction of arrival. Face detection is done on the Nao probably using Viola-Jones face detection method.The algorithm implemented in Matlab for Nao is presented in Algorithm 3 and summarized in Figure 5.2.

Algorithm 3 Audio Visual Speaker Loacalization with Nao

Require: Connection to Nao Robot
procedure SPEAKERLOCALIZATION(audio,video)
    Learn background noise statistics
    repeat
        Grab audio signal from Nao
        Grab image from Nao
        if Sound detected **then**
            Remove background noise
            $x \leftarrow$ Direction of sound
            if x in image **then**
                Detect face close to sound direction
                Move Nao's head to center the face
            else
                Move Nao's head
                Grab new image
                Detect face close to sound direction
                Move Nao's head to center the face
            end if
        end if
    until True
end procedure

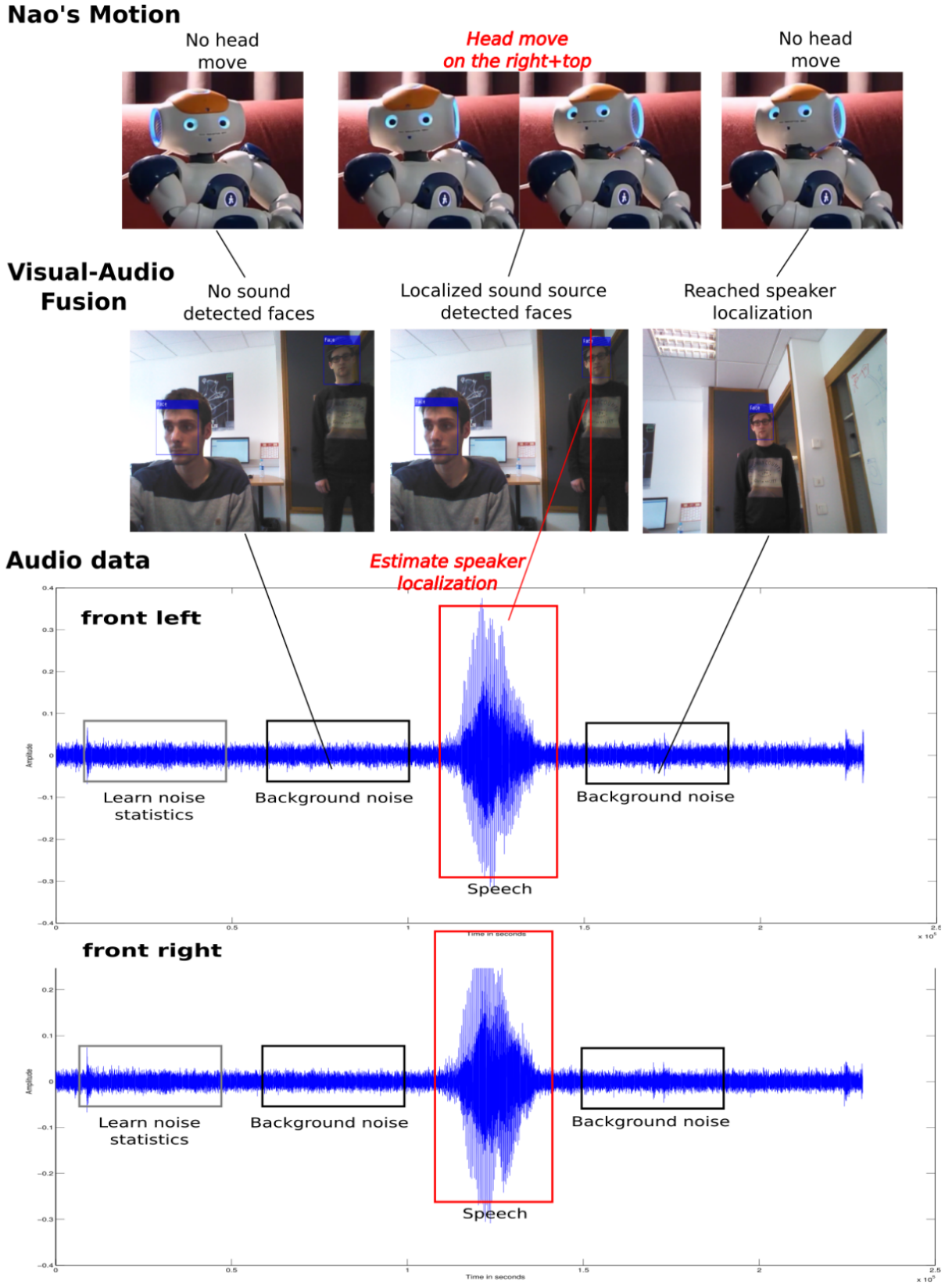

Figure 5.2: Speaker localization module on the Nao robot

5.3 CONCLUSION

This module was the first application of the NaoLab project. Later Perception team members improved and extended the method with more robust sound localization method and added a multiperson tracker to keep track of all the people in the scene around Nao. A publication came out of this project at the International Conference on Multimodal Interaction in 2015, [Badeig 15], the paper was accepted for a demonstration where this algorithm was presented. More details can be found at the following link *team.inria.fr/perception/naolab-toolbox/*.

CHAPTER 6

CONCLUSION

6.1 SUMMARY AND DISCUSSION

In this chapter I would like to go back through all the work done and presented in this thesis, from the starting idea 3 years ago to the final results obtained some days ago. We started with my supervisor Radu Horaud on the idea that a mixture of linear regression, inspired by [Deleforge 15] and [Deleforge 14], could model the relation between the head pose and the visual representation of a face.

Under Radu's guidance we investigated how this model behaves for head-pose estimation, how the parameters influence the estimation accuracy under realistic conditions. We ended up with an algorithm to estimate jointly the head pose and the possible face bounding-box misalignment due to face detection. Experiments showed that the head-pose estimation gives better results than most of regression methods and similar or better than state-of-the-art methods.

After achieving this step, we decided to extend the regression model for two reasons, first we wanted to improve the accuracy and we had the belief that temporal models will manage this, the addition of information from the past will lead in a more accurate estimation. Secondly a temporal model could smooth the estimation output and correct some inaccurate estimations due to errors from the input. We embeded the mixture of affine regressions into a Kalman filter framework and ended up with a switching dynamic model. The issue with switching dynamic model is that the number of components in their final mixture model increases exponentially at each time step. A solution for this is to use the merging moments method, the GPB2 that combines the components at each step, by merging components moments which value of Z diverges at time $t-1$. This merging of components keeps the number of components in the mixture fixed to K at each time step. Unfortunately using the GPB2 approximation arises a new problem, the estimation was too slow to work real time for head-pose estimation problem. Indeed at each time step, parameters of K^2+K Gaussian distributions have to be estimated making it difficult for high enough values of K to work in *real time*. Thus we looked a bit at the literature

to see the alternative, variational approximation came out. We defined an approximation that breaks the time dependencies between the variables, thus avoiding any increase of the number of components in the mixture. Experiments revealed that these two tracking methods achieved better results than the regression method and than tracking with a Kalman filter.

These two contributions presented separatly in the manuscript are working into the same unified framework in order to solve the problem of head-pose tracking for human robot interaction.

6.2 Direction for Future Research

This is not the end, we would like to see the work done in this thesis as an opening to new research, we defined a non-exhaustive list of the possibility of new directions from this work:

- A nice direction to explore would be to look more in the geometry of the low dimensional space. Transforming the parameter $\mathbf{C}$ into a temporal variable of the model, that would depend on Z_t, $\boldsymbol{x}_{t-1}$ and the velocity of $\boldsymbol{x}_t$. Adding this to the model could help to improve the prediction part of the model to obtain more accurate and smoother estimation.
- Using the strength of deep learning and convolutional neural networks model to provide good feature representation and also good accuracy for regression. Combining a CNN for head-pose estimation with a tracking algorithm of the type as the ones presented in this thesis or developing a recurrent neural networks for head pose.
- Staying in the robot perception by combining the head-pose tracking and the sound source tracking into a general framework for scene analysis and then jump to interaction step and the decision making, using the perception as the observation of scene. Combining the head-pose estimation and the sound source tracking in a single framework.
- To go to the next level, in the thesis we focus on the perception, but now the next step is the interaction, the communication being able from head pose speaker position and other cues to determine which action the robot has to do. Start a conversation if a person is looking at the robot, call a person to look at it, to look at the focus of attention of the scene.

APPENDIX A

This appendix details the EM algorithm that estimates the parameters $\boldsymbol{\theta}$ of the regression method described in Chapter 2. The algorithm is separated in 2 steps: the E step serves to estimate the posterior distribution of the latent variables of the model and the M step that updates the parameters of the model $\boldsymbol{\phi}$. Once initialized, at each iteration i, the algorithm alternates between the E-step and the M-step.

E-STEP

For the hybrid GLLiM (hGLLiM), the E-step is split into two expection steps, the E-W-step computes the posterior of the latent part $\boldsymbol{x}$ and the E-Z-step computes the posterior of the assignment latent variable Z. Given a training set of pairs $\{\boldsymbol{t}_n, \boldsymbol{y}_n\}_{n=1}^N$, $\boldsymbol{t}$ the target and $\boldsymbol{y}$ the observation.

E-W-STEP

Given the current parameter estimates $\boldsymbol{\theta}$, the posterior probability is fully determined by the distributions $p(\boldsymbol{w}_n|Z_n = k, \boldsymbol{t}_n, \boldsymbol{y}_n; \boldsymbol{\theta})$ for all n and k, which can be shown to be Gaussian. Their covariance matrices $\mathbf{S}_k^w$ and vector means $\boldsymbol{\mu}_{nk}^w$ are given by

$$\mathbf{S}_k^w = \left(\mathbf{I} + \mathbf{A}_k^{w\top} \boldsymbol{\Sigma}_k^{-1} \mathbf{A}_k^w\right)^{-1}, \tag{1}$$

$$\boldsymbol{\mu}_{nk}^w = \mathbf{S}_k^w \mathbf{A}_k^{w\top} \boldsymbol{\Sigma}_k^{-1} \left(\boldsymbol{y}_n - \mathbf{A}_k^t \boldsymbol{t}_n - \boldsymbol{b}_k\right). \tag{2}$$

E-Z-STEP

The posterior of Z_n is determined by:

$$r_{nk} = p(Z_n = k|\boldsymbol{t}_n, \boldsymbol{y}_n; \boldsymbol{\theta}) = \frac{\pi_k p(\boldsymbol{y}_n, \boldsymbol{t}_n|Z_n = k; \boldsymbol{\theta})}{\sum_{j=1}^K \pi_j p(\boldsymbol{y}_n, \boldsymbol{t}_n|Z_n = j; \boldsymbol{\theta})} \tag{3}$$

for all n and k, where

$$p(\boldsymbol{y}_n, \boldsymbol{t}_n|Z_n = k; \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{t}_n; \boldsymbol{c}_k^t, \boldsymbol{\Gamma}_k^t)\mathcal{N}(\boldsymbol{y}_n; \boldsymbol{d}_k, \boldsymbol{\Phi}_k), \tag{4}$$

with:

$$\boldsymbol{d}_k = \mathbf{A}_k^t \boldsymbol{t}_n + \boldsymbol{b}_k,$$
$$\boldsymbol{\Phi}_k = \mathbf{A}_k^w \mathbf{A}_k^{w\top} + \boldsymbol{\Sigma}_k.$$

The maximization can then be performed using the posterior probabilities $p(Z_n = k|\boldsymbol{t}_n, \boldsymbol{y}_n; \boldsymbol{\theta})$ and its parameters $\boldsymbol{\mu}_{nk}^w$ and $\mathbf{S}_k^w$. We use the following notations: $\rho_{nk} = r_{nk} / \sum_{n=1}^N r_{nk}$ and $\boldsymbol{x}_{nk} = [\boldsymbol{t}_n; \boldsymbol{\mu}_{nk}^w] \in \mathbb{R}^L$.

M-STEP

The M-step is also divided in two parts, the M-GMM-step updates the parameters of the model related to $\boldsymbol{x}$ and the M-Mapping-step updates the parameters of the affine transformations.

M-GMM-STEP

The updating of parameters $\pi_k, \boldsymbol{c}_k^t$ and $\boldsymbol{\Gamma}_k^t$ correspond to those of a standard Gaussian mixture model on $\boldsymbol{T}_{1:N}$, so that we get straightforwardly:

$$\boldsymbol{c}_k^t = \sum_{n=1}^N \rho_{nk}^{(i)} \boldsymbol{t}_n, \tag{5}$$

$$\boldsymbol{\Gamma}_k^t = \sum_{n=1}^N \rho_{nk} (\boldsymbol{t}_n - \boldsymbol{c}_k^t)(\boldsymbol{t}_n - \boldsymbol{c}_k^t)^\top \tag{6}$$

$$\pi_k = \frac{\sum_{n=1}^N r_{nk}}{N}. \tag{7}$$

M-MAPPING-STEP

The updating of mapping parameters $\{\mathbf{A}_k, \boldsymbol{b}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ is also in closed-form. The affine transformation matrix is updated with:

$$\mathbf{A}_k = \mathbf{Y}_k \mathbf{X}_k{}^\top (\mathbf{S}_k^{\mathrm{x}} + \mathbf{X}_k \mathbf{X}_k{}^\top)^{-1} \tag{8}$$

where:

$$
\begin{aligned}
\mathbf{X}_k &= \left(\sqrt{\rho_{1k}}(\boldsymbol{x}_{1k} - \boldsymbol{x}_k), \ldots, \sqrt{\rho_{nk}}(\boldsymbol{x}_{Nk} - \boldsymbol{x}_k)\right), \\
\mathbf{Y}_k &= \left(\sqrt{\rho_{1k}}(\boldsymbol{y}_1 - \boldsymbol{y}_k), \ldots, \sqrt{\rho_{nk}}(\boldsymbol{y}_N - \boldsymbol{y}_k)\right), \\
\boldsymbol{x}_k &= \sum_{n=1}^{N} \rho_{nk} \boldsymbol{x}_{nk}, \\
\boldsymbol{y}_k &= \sum_{n=1}^{N} \rho_{nk} \boldsymbol{y}_n, \\
\mathbf{S}_k^{\mathrm{x}} &= \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_k^w \end{pmatrix}.
\end{aligned}
$$

The intercept parameters are updated with:

$$
\boldsymbol{b}_k = \sum_{n=1}^{N} \rho_{nk} (\boldsymbol{y}_n - \mathbf{A}_k \boldsymbol{x}_{nk}). \tag{9}
$$

The noise covariance matrices are updated with:

$$
\begin{aligned}
\boldsymbol{\Sigma}_k = \operatorname{diag}\Big\{ & \mathbf{A}_k^w \mathbf{S}_k^w \mathbf{A}_k^{w\top} + \\
& \sum_{n=1}^{N} \rho_{nk} (\boldsymbol{y}_n - \mathbf{A}_k \boldsymbol{x}_{nk} - \boldsymbol{b}_k)(\boldsymbol{y}_n - \mathbf{A}_k \boldsymbol{x}_{nk} - \boldsymbol{b}_k)^\top \Big\}
\end{aligned} \tag{10}
$$

where the $\operatorname{diag}\{\cdot\}$ operator sets all the off-diagonal entries to 0. Initial parameters $\boldsymbol{\theta}$ are obtained by fitting a GMM with K components to the joint output-input training dataset $\{\boldsymbol{t}_n, \boldsymbol{y}_n\}_{n=1}^N$.

APPENDIX B

This appendix details the derivation of the EM algorithm that estimates the parameters ϕ of the temporal model presented in Chapter 3. The algorithm is separated in 2 steps: the E step serves to estimate the posterior of the latent variables, in the case of the model using GPB2 approximation, this is derived in Chapter 3. Therefore here we only derived the E step for the variational approximation. The M step is derived for both approximations.

E STEP

The E step of the algorithm computes the posterior distribution of the latent variables $\boldsymbol{x}_t$ and Z_t. The E step is separated in two parts, E-Z step that compute the posterior distributions of Z and E-X step that do the same for X. The posterior distribution are obtained using the following formulas:

$$\log q^*(Z_t) = \mathbb{E}_{q(\boldsymbol{x}_{1:T}, Z_{1:T} \setminus Z_t)} \left[\log p\left(\boldsymbol{x}_{1:T}, Z_{1:T}, \boldsymbol{y}_{1:T}\right)\right] \tag{11}$$

$$\log q^*(\boldsymbol{x}_t) = \mathbb{E}_{q(\boldsymbol{x}_{1:T}, Z_{1:T} \setminus \boldsymbol{x}_t)} \left[\log p\left(\boldsymbol{x}_{1:T}, Z_{1:T}, \boldsymbol{y}_{1:T}\right)\right] \tag{12}$$

with the loglikelihood $\log p\left(\boldsymbol{x}_{1:T}, Z_{1:T}, \boldsymbol{y}_{1:T}\right)$ defined as:

$$\log p\left(\boldsymbol{x}_{1:T}, Z_{1:T}, \boldsymbol{y}_{1:T}\right) \approx \log \left[\prod_{t=2}^{T} p(\boldsymbol{y}_t|\boldsymbol{x}_t, Z_t) p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, Z_t) p(Z_t|Z_{t-1}) \right] \tag{13}$$

$$\approx \log \left[\prod_{t=2}^{T} \prod_{j=1}^{K} \Bigg(p(\boldsymbol{y}_t|\boldsymbol{x}_t, Z_t = j) p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, Z_t = j) \times \prod_{i=1}^{K} p(Z_t = j|Z_{t-1} = i)^{\alpha_{t-1,i}} \Bigg)^{\alpha_{t,j}} \right] \tag{14}$$

$$\approx \sum_{t=2}^{T} \sum_{j=1}^{K} \alpha_{t,j} \Bigg(\log p(\boldsymbol{y}_t|\boldsymbol{x}_t, Z_t = j) + \log p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, Z_t = j) \times \sum_{i=1}^{K} \alpha_{t-1,i} \log p(Z_t = j|Z_{t-1} = i) \Bigg) \tag{15}$$

E-Z STEP

Replacing the loglikelihood in Equation (11) by its true value and keeping only the term that contains Z_t, and using that $q\left(\boldsymbol{X}, Z \setminus Z_t\right) = \prod_{t=1}^{T} q(\boldsymbol{x}_t) \prod_{t=1}^{T} q(Z_t)$ the optimal log posterior distribution of Z_t becomes:

$$\begin{aligned}\log q^*\left(Z_t\right) \approx & \mathbb{E}_{q(\boldsymbol{x}_t)}\left[\log p(\boldsymbol{y}_t|\boldsymbol{x}_t, Z_t = j)\right] + \mathbb{E}_{q(\boldsymbol{x}_t)q(\boldsymbol{x}_{t-1})}\left[\log p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, Z_t = j)\right] \\ & + \sum_{i=1}^{K} \mathbb{E}_{q(Z_{t-1})}\left[\alpha_{t-1,i} \log p(Z_t|Z_{t-1})\right] + \sum_{i=1}^{K} \mathbb{E}_{q(Z_{t+1})}\left[\alpha_{t+1,i} \log p(Z_{t+1}|Z_t)\right]\Bigg].\end{aligned} \tag{16}$$

$$\begin{aligned}\mathbb{E}_{q(\boldsymbol{x}_t)}\left[\log p(\boldsymbol{y}_t|\boldsymbol{x}_t, Z_t = j)\right] = & \mathbb{E}_{q(\boldsymbol{x}_t)}\Bigg[-\frac{1}{2}\left(\log|2\pi\boldsymbol{\Sigma}_j|\right. \\ & \left. + \left(\boldsymbol{y}_t - \mathbf{A}_j\boldsymbol{x}_t - \boldsymbol{b}_j\right)^\top \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{y}_t - \mathbf{A}_j\boldsymbol{x}_t - \boldsymbol{b}_j\right)\right)\Bigg] \end{aligned} \tag{17}$$

$$\begin{aligned} = & -\frac{1}{2}\left(\log|2\pi\boldsymbol{\Sigma}_j|\right. \\ & \left. + \mathbb{E}_{q(\boldsymbol{x}_t)}\left[\left(\boldsymbol{y}_t - \mathbf{A}_j\boldsymbol{x}_t - \boldsymbol{b}_j\right)^\top \boldsymbol{\Sigma}_j^{-1}\left(\boldsymbol{y}_t - \mathbf{A}_j\boldsymbol{x}_t - \boldsymbol{b}_j\right)\right]\right) \end{aligned} \tag{18}$$

$$\begin{aligned} = & -\frac{1}{2}\left(\log|2\pi\boldsymbol{\Sigma}_j|\right. \\ & + \left(\boldsymbol{y}_t - \mathbf{A}_j\mathbb{E}_{q(\boldsymbol{x}_t)}\left[\boldsymbol{x}_t\right] - \boldsymbol{b}_j\right)^\top \boldsymbol{\Sigma}_j^{-1}\left(\boldsymbol{y}_t - \mathbf{A}_j\mathbb{E}_{q(\boldsymbol{x}_t)}\left[\boldsymbol{x}_t\right] - \boldsymbol{b}_j\right) \\ & \left. + \mathrm{Tr}\left[\mathbf{A}_j^\top \boldsymbol{\Sigma}_j^{-1}\mathbf{A}_j \mathrm{Cov}(\boldsymbol{x}_t)\right]\right) \end{aligned} \tag{19}$$

$$\begin{aligned} & \mathbb{E}_{q(\boldsymbol{x}_t)}\left[\mathbb{E}_{q(\boldsymbol{x}_{t-1})}\left[\log p(\boldsymbol{c}_t|\boldsymbol{x}_{t-1}, Z_t = j)\right]\right] \\ & = \mathbb{E}_{q(\boldsymbol{x}_t)}\left[\mathbb{E}_{q(\boldsymbol{x}_t)}\left[-\frac{1}{2}\left(\log|2\pi\mathbf{Q}_j| + \left(\boldsymbol{x}_t - \mathbf{C}_j\boldsymbol{x}_{t-1}\right)^\top \mathbf{Q}^{-1}\left(\boldsymbol{x}_t - \mathbf{C}_j\boldsymbol{x}_{t-1}\right)\right)\right]\right] \end{aligned} \tag{20}$$

$$\begin{aligned} & = \mathbb{E}_{q(\boldsymbol{x}_t)}\left[-\frac{1}{2}\left(\log|2\pi\mathbf{Q}_j| + \left(\boldsymbol{x}_t - \mathbf{C}_j\mathbb{E}_{q(\boldsymbol{x}_{t-1})}\left[\boldsymbol{x}_{t-1}\right]\right)^\top \mathbf{Q}^{-1}\left(\boldsymbol{x}_t - \mathbf{C}_j\mathbb{E}_{q(\boldsymbol{x}_{t-1})}\left[\boldsymbol{x}_{t-1}\right]\right)\right.\right. \\ & \quad \left.\left. + \mathrm{Tr}\left[\mathbf{C}_j^\top \mathbf{Q}_j^{-1}\mathbf{C}_j \mathrm{Cov}(\boldsymbol{x}_{t-1})\right]\right)\right] \end{aligned} \tag{21}$$

$$\begin{aligned} & = -\frac{1}{2}\left(\log|2\pi\mathbf{Q}_j|\right. \\ & \quad + \left(\mathbb{E}_{q(\boldsymbol{x}_t)}\left[\boldsymbol{x}_t\right] - \mathbf{C}_j\mathbb{E}_{q(\boldsymbol{x}_{t-1})}\left[\boldsymbol{x}_{t-1}\right]\right)^\top \mathbf{Q}^{-1}\left(\mathbb{E}_{q(\boldsymbol{x}_t)}\left[\boldsymbol{x}_t\right] - \mathbf{C}_j\mathbb{E}_{q(\boldsymbol{x}_{t-1})}\left[\boldsymbol{x}_{t-1}\right]\right) \\ & \quad \left. + \mathrm{Tr}\left[\mathbf{C}_j^\top \mathbf{Q}_j^{-1}\mathbf{C}_j \mathrm{Cov}(\boldsymbol{x}_{t-1})\right] + \mathrm{Tr}\left[\mathbf{Q}_j^{-1}\mathrm{Cov}(\boldsymbol{x}_t)\right]\right) \end{aligned} \tag{22}$$

$$\begin{aligned} \mathbb{E}_{q(Z_{t-1})}\left[\alpha_{t-1,i} \log p(Z_t = j|Z_{t-1} = i)\right] = & \mathbb{E}_{q(Z_{t-1})}\left[\alpha_{t-1,i}\right] \log p(Z_t = j|Z_{t-1} = i) \quad (23) \\ = & q^*(Z_{t-1} = i) \log \tau_{ij} \quad (24) \end{aligned}$$

$$\mathbb{E}_{q(Z_{t+1})}\left[\alpha_{t+1,i}\log p(Z_{t+1}=i|Z_t=j)\right] = \mathbb{E}_{q(Z_{t+1})}\left[\alpha_{t+1,i}\right]\log p(Z_{t+1}=i|Z_t=j) \quad (25)$$
$$= q^*(Z_{t+1}=i)\log\tau_{ji} \quad (26)$$

Therefore the posterior distribution of Z_t can be finally written as:

$$\begin{aligned} q^*\left(Z_t\right) = \prod_{j=1}^{K}\Bigg[& \mathcal{N}\left(\boldsymbol{y}_t;\mathbf{A}_j\mathbb{E}[\boldsymbol{x}_t]+\boldsymbol{b}_j,\boldsymbol{\Sigma}_j\right)\exp\left(-\frac{1}{2}\mathrm{tr}\left[\mathbf{A}_j^\top\boldsymbol{\Sigma}_j^{-1}\mathbf{A}_j\mathrm{Cov}(\boldsymbol{x}_t)\right]\right) \\ & \times\mathcal{N}\left(\mathbb{E}[\boldsymbol{x}_t];\mathbf{C}_j\mathbb{E}[\boldsymbol{x}_{t-1}],\mathbf{Q}_j\right)\exp\left(-\frac{1}{2}\mathrm{tr}\left[\mathbf{C}_j^\top\mathbf{Q}_j^{-1}\mathbf{C}_j\mathrm{Cov}(\boldsymbol{x}_{t-1})\right]\right) \\ & \times\exp\left(-\frac{1}{2}\mathrm{tr}\left[\mathbf{Q}_j^{-1}\mathrm{Cov}(\boldsymbol{x}_t)\right]\right)\prod_{j=1}^{K}\tau_{ij}^{q^*(Z_{t-1}=i)}\prod_{j=1}^{K}\tau_{ji}^{q^*(Z_{t+1}=i)}\Bigg]^{\alpha_{t,j}}. \end{aligned} \quad (27)$$

Which allows us to express $q^*\left(Z_t=j\right)$ as follows:

$$q^*\left(Z_t=j\right)=\frac{q^*\left(Z_t|\alpha_{t,j}=1\right)}{\sum_{i=1}^{K}q^*\left(Z_t|\alpha_{t,i}=1\right)}. \quad (28)$$

E-X STEP

The derivations to obtain the log posterior distribution of $\boldsymbol{x}_t$ follow the same principle as for Z_t,

$$\begin{aligned} \log q^*\left(\boldsymbol{x}_t\right) \approx & \sum_{j=1}^{K}\mathbb{E}_{q(Z_t)}\left[\alpha_{t,j}\right]\left[\log p(\boldsymbol{y}_t|\boldsymbol{x}_t,Z_t=j)+\mathbb{E}_{q(\boldsymbol{x}_{t-1})}\left[\log p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1},Z_t=j)\right]\right] \\ & +\sum_{j=1}^{K}\mathbb{E}_{q(Z_{t+1})}\left[\alpha_{t+1,j}\right]\left[\mathbb{E}_{q(\boldsymbol{x}_{t+1})}\left[\log p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t,Z_{t+1}=j)\right]\right] \end{aligned} \quad (29)$$

$$\mathbb{E}_{q(Z_t)}\left[\alpha_{t,j}\right]=q^*(Z_t=j) \quad (30)$$
$$\mathbb{E}_{q(Z_{t+1})}\left[\alpha_{t+1,j}\right]=q^*(Z_{t+1}=i) \quad (31)$$

$$\begin{aligned} & \mathbb{E}_{q(\boldsymbol{x}_{t-1})}\left[\log p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1},Z_t=j)\right] \\ & =-\frac{1}{2}\Bigg(\log|2\pi\mathbf{Q}_j|+\left(\boldsymbol{x}_t-\mathbf{C}_j\mathbb{E}_{q(\boldsymbol{x}_{t-1})}\left[\boldsymbol{x}_{t-1}\right]\right)^\top\mathbf{Q}_j^{-1}\left(\boldsymbol{x}_t-\mathbf{C}_j\mathbb{E}_{q(\boldsymbol{x}_{t-1})}\left[\boldsymbol{x}_{t-1}\right]\right) \\ & \quad +\mathrm{Tr}\left[\mathbf{C}_j^\top\mathbf{Q}_j^{-1}\mathbf{C}_j\mathrm{Cov}(\boldsymbol{x}_{t-1})\right]\Bigg) \end{aligned} \quad (32)$$

Likewise:

$$\begin{aligned}
&\mathbb{E}_{q(\boldsymbol{x}_{t+1})}\left[\log p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, Z_{t+1}=i)\right] \\
&\quad = -\frac{1}{2}\Bigg(\log|2\pi\mathbf{Q}_i| + \left(\mathbb{E}_{q(\boldsymbol{x}_{t+1})}\left[\boldsymbol{x}_{t+1}\right] - \mathbf{C}_i\boldsymbol{x}_t\right)^\top \mathbf{Q}_i^{-1}\left(\mathbb{E}_{q(\boldsymbol{x}_{t+1})}\left[\boldsymbol{x}_{t+1}\right] - \mathbf{C}_i\boldsymbol{x}_t\right) \\
&\qquad + \mathrm{Tr}\left[\mathbf{Q}_i^{-1}\mathrm{Cov}(\boldsymbol{x}_{t+1})\right]\Bigg)
\end{aligned} \tag{33}$$

Therefore

$$\begin{aligned}
\log q^*\left(\boldsymbol{x}_t\right) \approx -\frac{1}{2}\Bigg[&\boldsymbol{x}_t^\top\left(\sum_{j=1}^K q^*(Z_t=j)\left(\mathbf{A}_j^\top\boldsymbol{\Sigma}_j^{-1}\mathbf{A}_j + \mathbf{Q}_j^{-1}\right) + \sum_{i=1}^K q^*(Z_{t+1}=i)\mathbf{C}_i^\top\mathbf{Q}_i^{-1}\mathbf{C}_i\right)\boldsymbol{x}_t \\
&- 2\boldsymbol{x}_t^\top\Bigg(\sum_{j=1}^K q^*(Z_t=j)\left(\mathbf{A}_j^\top\boldsymbol{\Sigma}_j^{-1}\left(\boldsymbol{y}_t - \boldsymbol{b}_j\right) + \mathbf{Q}_j^{-1}\mathbf{C}_j\mathbb{E}_{q(\boldsymbol{x}_{t-1})}\left[\boldsymbol{x}_{t-1}\right]\right) \\
&+ \sum_{i=1}^K q^*(Z_{t+1}=i)\mathbf{C}_i^\top\mathbf{Q}_i^{-1}\mathbb{E}_{q(\boldsymbol{x}_{t+1})}\left[\boldsymbol{x}_{t+1}\right]\Bigg) \\
&+ \sum_{j=1}^K q^*(Z_t=i)\Big(\log|\mathbf{Q}_j| + \log|\boldsymbol{\Sigma}_j| + (\boldsymbol{y}_t-\boldsymbol{b}_j)^\top\boldsymbol{\Sigma}_j^{-1}(\boldsymbol{y}_t-\boldsymbol{b}_j) \\
&+ \mathbb{E}_{q(\boldsymbol{x}_{t-1})}\left[\boldsymbol{x}_{t-1}\right]^\top\mathbf{C}_j^\top\mathbf{Q}_j^{-1}\mathbf{C}_j\mathbb{E}_{q(\boldsymbol{x}_{t-1})}\left[\boldsymbol{x}_{t-1}\right]\Big) \\
&+ \sum_{i=1}^K q^*(Z_{t+1}=i)\left(\log|\mathbf{Q}_i| + \mathbb{E}_{q(\boldsymbol{x}_{t+1})}\left[\boldsymbol{x}_{t+1}\right]^\top\mathbf{Q}_i^{-1}\mathbb{E}_{q(\boldsymbol{x}_{t+1})}\left[\boldsymbol{x}_{t+1}\right]\right)\Bigg]
\end{aligned} \tag{34}$$

$$q^*\left(\boldsymbol{x}_t\right) \propto \mathcal{N}\left(\boldsymbol{x}_t; \boldsymbol{\eta}_t, \mathbf{V}_t\right) \tag{35}$$

with the parameters of the distribution expressed as:

$$\mathbf{V}_t = \left[\sum_{j=1}^K \mathbb{E}[\alpha_{t,j}]\left(\mathbf{A}_j^\top\boldsymbol{\Sigma}_j^{-1}\mathbf{A}_j + \mathbf{Q}_j^{-1}\right) + \sum_{i=1}^K \mathbb{E}[\alpha_{t+1,i}]\mathbf{C}_i^\top\mathbf{Q}_i^{-1}\mathbf{C}_i\right]^{-1} \tag{36}$$

$$\begin{aligned}
\boldsymbol{\eta}_t = &\mathbf{V}_t\Bigg[\sum_{j=1}^K \mathbb{E}[\alpha_{t,j}]\left(\mathbf{A}_j^\top\boldsymbol{\Sigma}_j^{-1}(\boldsymbol{y}_t - \boldsymbol{b}_t) + \mathbf{Q}_j^{-1}\mathbf{C}_j\mathbb{E}[\boldsymbol{x}_{t-1}]\right) \\
&+ \sum_{i=1}^K \mathbb{E}[\alpha_{t+1,i}]\mathbf{C}_i^\top\mathbf{Q}_i^{-1}\mathbb{E}[\boldsymbol{x}_{t+1}]\Bigg].
\end{aligned} \tag{37}$$

M STEP

The parameters ϕ are updated by maximizing the expected log likelihood for each parameter:

$$\boldsymbol{\phi}^{new} = \underset{\boldsymbol{\phi}}{\operatorname{argmax}}(\mathbb{E}_{p(\boldsymbol{x}_{1:T}, Z_{1:T}|\boldsymbol{y}_{1:T})}[\mathcal{L}]) \tag{38}$$

where $\mathcal{L}$ represents the complete data loglikelihood. Maximizing the expected log likelihood is done by solving:

$$\frac{\partial \mathbb{E}_{p(\boldsymbol{x}_{1:T}, Z_{1:T}|\boldsymbol{y}_{1:T})}[\mathcal{L}]}{\partial \phi} = 0. \tag{39}$$

The complete data loglikelihood is defined as follows:

$$\begin{aligned}\mathcal{L} &= \log p\left(\boldsymbol{x}_{1:T}, Z_{1:T}, \boldsymbol{y}_{1:T}; \theta\right) \\ &= \sum_{t=1}^{T} \log \mathcal{N}(\boldsymbol{y}_t; \mathbf{A}_{Z_t}\boldsymbol{x}_t + \boldsymbol{b}_{Z_t}, \boldsymbol{\Sigma}_{Z_t}) + \sum_{t=2}^{T} \log \mathcal{N}(\boldsymbol{x}_t; \mathbf{C}_{Z_t}\boldsymbol{x}_{t-1}, \mathbf{Q}_{Z_t}) \\ &\quad + \sum_{t=2}^{T} \log \tau_{Z_{t-1}, Z_t} + \log p\left(\boldsymbol{x}_1, Z_1\right).\end{aligned} \tag{40}$$

The posterior distribution of $\boldsymbol{x}_t$ and Z_t being different for each approximation, the numerator in Equation (3.72) expression varies between the GPB2 and variational approximation, when one is using the GPB2 approximation the expected loglikelihood is defined as:

$$\mathbb{E}_{p(\boldsymbol{x}_{1:T}, Z_{1:T}|\boldsymbol{y}_{1:T})}[\mathcal{L}] \approx \mathbb{E}_{p(Z_{1:T}|\boldsymbol{y}_{1:T})}\left[\mathbb{E}_{p(\boldsymbol{x}_{1:T}|Z_{1:T}\boldsymbol{y}_{1:T})}[\mathcal{L}]\right]. \tag{41}$$

and for the variational approximation:

$$\mathbb{E}_{p(\boldsymbol{x}_{1:T}, Z_{1:T}|\boldsymbol{y}_{1:T})}[\mathcal{L}] \approx \mathbb{E}_{\prod_{t=1}^{T} q^*(Z_t)}\left[\mathbb{E}_{\prod_{t=1}^{T} q^*(\boldsymbol{x}_t)}[\mathcal{L}]\right]. \tag{42}$$

ESTIMATION OF $\mathbf{C}_j$

The update formula for $\mathbf{C}_j$ is obtained in 3 steps. First by expressing the part of the expected log likelihood that contains $\mathbf{C}_j$:

$$\begin{aligned}&\mathbb{E}_{p(\boldsymbol{x}_{1:T}, Z_{1:T}|\boldsymbol{y}_{1:T})}\left[\mathcal{L}_{\mathbf{C}_j}\right] \\ &\quad \approx \mathbb{E}_{p(Z_{1:T}|\boldsymbol{y}_{1:T})}\left[\mathbb{E}_{p(\boldsymbol{x}_{1:T}|Z_{1:T}, \boldsymbol{y}_{1:T})}\left[-\sum_{t=2}^{T}\frac{1}{2}\left(\boldsymbol{x}_t - \mathbf{C}_j\boldsymbol{x}_{t-1}\right)^{\top}\mathbf{Q}_j^{-1}\left(\boldsymbol{x}_t - \mathbf{C}_j\boldsymbol{x}_{t-1}\right)\right]\right] \\ &\quad \approx -\frac{1}{2}\sum_{t=2}^{T} p(Z_t = j|\boldsymbol{y}_{1:T})\mathbb{E}_{p(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}|Z_t = j\boldsymbol{y}_{1:T})}\left[\left(\boldsymbol{x}_t - \mathbf{C}_j\boldsymbol{x}_{t-1}\right)^{\top}\mathbf{Q}_j^{-1}\left(\boldsymbol{x}_t - \mathbf{C}_j\boldsymbol{x}_{t-1}\right)\right].\end{aligned} \tag{43}$$

Then by deriving it with respect to the parameter $\mathbf{C}_j$:

$$\begin{aligned}\frac{\partial \mathbb{E}\left[\mathcal{L}_{\mathbf{C}_j}\right]}{\partial \mathbf{C}_j} &\approx \sum_{t=2}^{T} p(Z_t = j|\boldsymbol{y}_{1:T})\mathbb{E}\left[\mathbf{Q}_j^{-1}\left(\boldsymbol{x}_t - \mathbf{C}_j\boldsymbol{x}_{t-1}\right)\boldsymbol{x}_{t-1}^{\top}\right] \\ &\approx \sum_{t=2}^{T} p(Z_t = j|\boldsymbol{y}_{1:T})\mathbf{Q}_j^{-1}\left(\mathbb{E}\left[\boldsymbol{x}_t\boldsymbol{x}_{t-1}^{\top}\right] - \mathbf{C}_j\mathbb{E}\left[\boldsymbol{x}_{t-1}\boldsymbol{x}_{t-1}^{\top}\right]\right) \end{aligned} \tag{44}$$

And finally by solving Equation (39) using previous equation:

$$\begin{aligned}\mathbf{C}_j \approx &\left(\sum_{t=2}^{T} p(Z_t = j|\boldsymbol{y}_{1:T})\mathbb{E}[\boldsymbol{x}_t\boldsymbol{x}_{t-1}^{\top}]\right) \\ &\times \left(\sum_{t=2}^{T} p(Z_t = j|\boldsymbol{y}_{1:T})\mathbb{E}[\boldsymbol{x}_{t-1}\boldsymbol{x}_{t-1}^{\top}]\right)^{-1},\end{aligned} \tag{45}$$

where:

$$\mathbb{E}\left[\boldsymbol{x}_t\boldsymbol{x}_{t-1}^{T}\right] = \mathbf{V}_{t,t-1} + \boldsymbol{\eta}_t\boldsymbol{\eta}_{t-1}^{\top}, \tag{46}$$

$$\mathbb{E}\left[\boldsymbol{x}_{t-1}\boldsymbol{x}_{t-1}^{T}\right] = \mathbf{V}_{t-1} + \boldsymbol{\eta}_{t-1}\boldsymbol{\eta}_{t-1}^{\top}, \tag{47}$$

$$\mathbb{E}\left[\boldsymbol{x}_t\boldsymbol{x}_t^{T}\right] = \mathbf{V}_t + \boldsymbol{\eta}_t\boldsymbol{\eta}_t^{\top} \tag{48}$$

The variational approximation by breaking the time dependency between $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t-1}$ simplifys the update formula for $\mathbf{C}_j$. Because $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t-1}$ are independent, $\mathbb{E}\left[\boldsymbol{x}_t\boldsymbol{x}_{t-1}^{T}\right]$ is equals to the outer product of the means. Therefore for the variational approximation the update formula for $\mathbf{C}_j$ is expressed as:

$$\begin{aligned}\mathbf{C}_j \approx &\left(\sum_{t=2}^{T} q^*\left(Z_t = j\right)\boldsymbol{\eta}_t\boldsymbol{\eta}_{t-1}^{\top}\right) \\ &\times \left(\sum_{t=2}^{T} q^*\left(Z_t = j\right)\boldsymbol{\eta}_{t-1}\boldsymbol{\eta}_{t-1}^{\top}\right)^{-1},\end{aligned} \tag{49}$$

ESTIMATION OF $\mathbf{Q}_j$

The derivations for the update formula of $\mathbf{Q}_j$ follow the same logic as the ones for $\mathbf{C}_j$, the expected log likelihood part that contains $\mathbf{Q}_j$ is:

$$\begin{aligned}
&\mathbb{E}_{p(\boldsymbol{x}_{1:T},Z_{1:T}|\boldsymbol{y}_{1:T})}\left[\mathcal{L}_{\mathbf{Q}_j}\right] \\
&\quad\approx\mathbb{E}_{p(Z_{1:T}|\boldsymbol{y}_{1:T})}\left[\mathbb{E}_{p(\boldsymbol{x}_{1:T}|Z_{1:T}\boldsymbol{y}_{1:T})}\left[-\sum_{t=2}^{T}\frac{1}{2}\left(\boldsymbol{x}_t-\mathbf{C}_j\boldsymbol{x}_{t-1}\right)^\top\mathbf{Q}_j^{-1}\left(\boldsymbol{x}_t-\mathbf{C}_j\boldsymbol{x}_{t-1}\right)\right]\right] \\
&\quad\approx-\frac{1}{2}\sum_{t=2}^{T}p(Z_t=j|\boldsymbol{y}_{1:T})\Bigg(\mathbb{E}_{p(\boldsymbol{x}_t,\boldsymbol{x}_{t-1}|Z_t=j\boldsymbol{y}_{1:T})}\left[\left(\boldsymbol{x}_t-\mathbf{C}_j\boldsymbol{x}_{t-1}\right)^\top\mathbf{Q}_j^{-1}\left(\boldsymbol{x}_t-\mathbf{C}_j\boldsymbol{x}_{t-1}\right)\right] \\
&\qquad+\log|\mathbf{Q}_j|\Bigg)
\end{aligned} \tag{50}$$

By deriving with respect to $\mathbf{Q}_j$ we obtain:

$$\frac{\partial\mathbb{E}\left[\mathcal{L}_{\mathbf{Q}_j}\right]}{\partial\mathbf{Q}_j^{-1}}\approx\sum_{t=2}^{T}p(Z_t=j|\boldsymbol{y}_{1:T})\Bigg(\mathbb{E}\left[\left(\boldsymbol{x}_t-\mathbf{C}_j\boldsymbol{x}_{t-1}\right)\left(\boldsymbol{x}_t-\mathbf{C}_j\boldsymbol{x}_{t-1}\right)^\top\right]-\mathbf{Q}_j\Bigg) \tag{51}$$

Finally the update formula is expressed as:

$$\begin{aligned}
\mathbf{Q}_j\approx&\frac{1}{\sum_{t=2}^{T}p(Z_t=j|\boldsymbol{y}_{1:T})} \\
&\times\left(\sum_{t=2}^{T}p(Z_t=j|\boldsymbol{y}_{1:T})\left(\mathbb{E}[\boldsymbol{x}_t\boldsymbol{x}_t^\top]-\mathbf{C}_j\mathbb{E}[\boldsymbol{x}_t\boldsymbol{x}_{t-1}^\top]\right)\right).
\end{aligned} \tag{52}$$

Likewise for $\mathbf{C}_j$, using the variational approximation the update formula becomes:

$$\begin{aligned}
\mathbf{Q}_j\approx&\frac{1}{\sum_{t=2}^{T}q^*\left(Z_t=j\right)} \\
&\times\left(\sum_{t=2}^{T}q^*\left(Z_t=j\right)\left(\boldsymbol{\eta}_t\boldsymbol{\eta}_t^\top-\mathbf{C}_j\boldsymbol{\eta}_t\boldsymbol{\eta}_{t-1}^\top\right)\right).
\end{aligned} \tag{53}$$

ESTIMATION OF τ_{ij}

The estimation of the transition parameter τ_{ij} follows the same logic as the one for $\mathbf{C}_j$ and $\mathbf{Q}_j$ but adds also a condition term:

$$\begin{aligned}
\tau_{ij}=\underset{\tau_{ij}}{\operatorname{argmax}}\,&\mathbb{E}_{p(\boldsymbol{x}_{1:T},Z_{1:T}|\boldsymbol{y}_{1:T})}\left[\mathcal{L}\right], \\
&\text{s.t.}\sum_{j=1}^{k}\tau_{ij}=1.
\end{aligned} \tag{54}$$

This is solved using the Lagrange multiplier, which consists of adding to the Expected loglikelihood $-\lambda\left(\sum_{j=1}^{K}\tau_{ij}-1\right)$, where λ is the Lagrange multiplier:

$$\mathbb{E}_{p(\boldsymbol{x}_{1:T},Z_{1:T}|\boldsymbol{y}_{1:T})}\left[\mathcal{L}_{\tau_{ij}}\right] \approx \sum_{t=2}^{T} p(Z_t=j|\boldsymbol{y}_{1:T})p(Z_{t-1}=i|\boldsymbol{y}_{1:T})\log\tau_{ij} - \lambda\left(\sum_{j=1}^{K}\tau_{ij}-1\right) \tag{55}$$

$$\frac{\partial\mathbb{E}\left[\mathcal{L}_{\tau_{ij}}\right]}{\partial\tau_{ij}} \approx \sum_{t=2}^{T}\frac{1}{\tau_{ij}}p(Z_t=j|\boldsymbol{y}_{1:T})p(Z_{t-1}=i|\boldsymbol{y}_{1:T}) - \lambda \tag{56}$$

Finally by solving Equation (39) for τ_{ij} with the Lagrange multiplier using the GPB2 approximation we obtain the following expression for the update of τ_{ij}:

$$\tau_{ij} \approx \frac{\sum_{t=2}^{T} p(Z_t=j|\boldsymbol{y}_{1:T})p(Z_{t-1}=i|\boldsymbol{y}_{1:T})}{\sum_{t=2}^{T} p(Z_{t-1}=i|\boldsymbol{y}_{1:T})} \tag{57}$$

And for the variational approximation:

$$\tau_{ij} \approx \frac{\sum_{t=2}^{T} q^*(Z_t=j)q^*(Z_{t-1}=i)}{\sum_{t=2}^{T} q^*(Z_{t-1}=i)} \tag{58}$$

PUBLICATIONS

INTERNATIONAL JOURNAL PUBLICATION

- [Drouard 17b] Vincent Drouard, Radu Horaud, Antoine Deleforge, Silèye Ba and Georgios Evangelidis. *Robust Head-Pose Estimation Based on Partially-Latent Mixture of Linear Regression*. In IEEE Transactions on Image Processing, Vol. 26, Issue 3, pages 1428-1440, March 2017.

INTERNATIONAL CONFERENCE PUBLICATIONS

- [Drouard 17a] Vincent Drouard, Silèye Ba and Radu Horaud. *Switching Linear Inverse-Regression Model for Tracking Head Pose*. In IEEE Winter Conference on Applications of Computer Vision (WACV), pages 1232 - 1240, Santa Rosa, CA, USA, March 2017.
- [Drouard 15] Vincent Drouard, Silèye Ba, Georgios Evangelidis, Antoine Deleforge and Radu Horaud. *Head pose estimation via probabilistic high-dimensional regression*. In IEEE International Conference on Image Processing (ICIP), pages 4624-4628, Quebec City, Canada, September 2015.
- [Badeig 15] Fabien Badeig, Quentin Pelorson, Soraya Arias, Vincent Drouard, Israel Gebru, Xiaofei Li, Georgios Evangelidis and Radu Horaud. *A distributed architecture for interacting with NAO*. In ACM on International Conference on Multimodal Interaction (ICMI), pages 385-386, Seattle, WA, USA, November 2015.
- [Deleforge 14] Antoine Deleforge, Vincent Drouard, Laurent Girin and Radu Horaud. *Mapping sounds onto images using binaural spectrograms*. In European Signal Processing Conference (EUSIPCO), pages 2470-2474, Lisbon, Portugal, September 2014.

REFERENCES

[Abdi 03] H. Abdi. *Partial least square regression (PLS regression)*. Encyclopedia for research methods for the social sciences, pages 792–795, 2003.

[Ahn 14] B. Ahn, J. Park & I. S. Kweon. *Real-time head orientation from a monocular camera using deep neural network*. In Asian Conference on Computer Vision, pages 82–96. Springer, 2014.

[Arulampalam 02] M. S. Arulampalam, S. Maskell, N. Gordon & T. Clapp. *A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking*. IEEE Transactions on Signal Processing, vol. 50, no. 2, pages 174–188, February 2002.

[Ba 04] S. O. Ba & J.-M. Odobez. *A probabilistic framework for joint head tracking and pose estimation*. In IEEE ICPR, August 2004.

[Badeig 15] F. Badeig, Q. Pelorson, S. Arias, V. Drouard, I. Gebru, X. Li, G. Evangelidis & R. Horaud. *A distributed architecture for interacting with NAO*. In ACM International Conference on Multimodal Interaction (ICMI), pages 385–386, November 2015.

[Baltrušaitis 16] T. Baltrušaitis, P. Robinson & L.-P. Morency. *Openface: an open source facial behavior analysis toolkit*. In IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2016.

[BenAbdelkader 10] C. BenAbdelkader. *Robust head pose estimation using supervised manifold learning*. In European Conference on Computer Vision, pages 518–531. Springer, 2010.

[Bhattacharyya 43] A. Bhattacharyya. *On a measure of divergence between two statistical population defined by their population distributions*. Bulletin Calcutta Mathematical Society, vol. 35, pages 99–109, July 1943.

[Bishop 07] C. Bishop. *Pattern Recognition and Machine Learning*, 2007.

[Deleforge 14] A. Deleforge, V. Drouard, L. Girin & R. Horaud. *Mapping sounds onto images using binaural spectrograms*. In European Signal Processing Conference (EUSIPCO), pages 2470–2474. IEEE, 2014.

[Deleforge 15] A. Deleforge, F. Forbes & R. Horaud. *High-Dimensional Regression with Gaussian Mixtures and Partially-Latent Response Variables*. Statistics and Computing, vol. 25, no. 5, pages 893–911, 2015.

[Demirkus 12] M. Demirkus, D. Precup, J. Clark & T. Arbel. *Soft biometric trait classification from real-world face videos conditioned on head pose estimation*. In IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 130–137, 2012.

[Demirkus 13] M. Demirkus, J. J. Clark & T. Arbel. *Robust semi-automatic head pose labeling for real-world face video sequences*. Multimedia Tools and Applications, vol. 70, no. 1, pages 495–523, 2013.

[Demirkus 14] M. Demirkus, D. Precup, J. J. Clark & T. Arbel. *Probabilistic Temporal Head Pose Estimation Using a Hierarchical Graphical Model*. In European Conference on Computer Vision, pages 328–344. 2014.

[Demirkus 15] M. Demirkus, D. Precup, J. J. Clark & T. Arbel. *Hierarchical temporal graphical model for head pose estimation and subsequent attribute classification in real-world videos*. Computer Vision and Image Understanding, vol. 136, pages 128 – 145, 2015. Generative Models in Computer Vision and Medical Imaging.

[Dornaika 04] F. Dornaika & F. Davoine. *Head and facial animation tracking using appearance-adaptive models and particle filters*. In IEEE CVPRW, June 2004.

[Drouard 15] V. Drouard, S. Ba, G. Evangelidis, A. Deleforge & R. Horaud. *Head pose estimation via probabilistic high-dimensional regression*. In IEEE International Conference on Image Processing (ICIP), pages 4624–4628, September 2015.

[Drouard 17a] V. Drouard, S. Ba & R. Horaud. *Switching Linear Inverse-Regression Model for Tracking Head Pose*. In IEEE Winter Conference on Applications of Computer Vision (WACV), pages 1232–1240, March 2017.

[Drouard 17b] V. Drouard, R. Horaud, A. Deleforge, S. Ba & G. Evangelidis. *Robust head-pose estimation based on partially-latent mixture of linear regressions*. IEEE Transactions on Image Processing, vol. 26, no. 3, pages 1428–1440, 2017.

[Fanelli 13] G. Fanelli, M. Dantone, J. Gall, A. Fossati & L. Van Gool. *Random Forests for Real Time 3D Face Analysis*. International Journal of Computer Vision, vol. 101, no. 3, pages 437–458, February 2013.

[Foytik 13] J. Foytik & V. Asari. *A Two-Layer Framework for Piecewise Linear Manifold-Based Head Pose Estimation*. International Journal of Computer Vision, vol. 101, no. 2, pages 270–287, January 2013.

[Gebru 17] I. D. Gebru, S. Ba, X. Li & R. Horaud. *Audio-Visual Speaker Diarization Based on Spatiotemporal Bayesian Fusion*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, 2017.

[Gee 96] A. Gee & R. Cipolla. *Fast visual tracking by temporal consensus*. Image and Vision Computing, vol. 14, no. 2, pages 105–114, March 1996.

[Ghahramani 96a] Z. Ghahramani & G. E. Hinton. *The EM algorithm for mixtures of factor analyzers*. Rapport technique CRG-TR-96-1, University of Toronto, 1996.

[Ghahramani 96b] Z. Ghahramani & G. E. Hinton. *Switching state-space models*. Rapport technique, Citeseer, 1996.

[Ghiass 15] R. S. Ghiass, O. Arandjelović & D. Laurendeau. *Highly accurate and fully automatic head pose estimation from a low quality consumer-level RGB-D sensor*. In Proceedings of the 2nd Workshop on Computational Models of Social Interactions: Human-Computer-Media Communication, pages 25–34. ACM, 2015.

[Gourier 04] N. Gourier, D. Hall & J. L. Crowley. *Estimating face orientation from robust detection of salient facial features*. In IEEE International Conference on Pattern Recognition Workshop on Visual Observation of Deictic Gestures, August 2004.

[Gourier 07] N. Gourier, J. Maisonnasse, D. Hall & J. L. Crowley. *Head pose estimation on low resolution images*. In Multimodal Technologies for Perception of Humans, pages 270–280. Springer, 2007.

[Haj 12] M. Haj, J. Gonzalez & L. Davis. *On partial least squares in head pose estimation: How to simultaneously deal with misalignment*. In IEEE Conference on Computer Vision and Pattern Recognition, pages 2602–2609, June 2012.

[Hu 05] N. Hu, W. Huang & S. Ranganath. *Head pose estimation by non-linear embedding and mapping*. In IEEE International Conference on Image Processing, volume 2, pages 342–345, September 2005.

[Jayagopi 12] D. B. Jayagopi, S. Sheikhi, D. Klotz, J. Wienke, J.-M. Odobez, S. Wrede, V. Khalidov, L. Nguyen, B. Wrede & D. Gatica-Perez. *The vernissage corpus: A multimodal human-robot-interaction dataset*. Rapport technique, 2012.

[Kooij 12] J. F. Kooij, G. Englebienne & D. M. Gavrila. *A non-parametric hierarchical model to discover behavior dynamics from tracks*. In ECCV. Springer, 2012.

[Li 07] Z. Li, Y. Fu, J. Yuan, T. Huang & Y. Wu. *Query Driven Localized Linear Discriminant Models for Head Pose Estimation*. In IEEE International Conference on Multimedia and Expo, pages 1810–1813, July 2007.

[Liu 16] X. Liu, W. Liang, Y. Wang, S. Li & M. Pei. *3D head pose estimation with convolutional neural network trained on synthetic images*. In IEEE International Conference on Image Processing, September 2016.

[Marin-Jimenez 14] M. Marin-Jimenez, A. Zisserman, M. Eichner & V. Ferrari. *Detecting People Looking at Each Other in Videos*. International Journal of Computer Vision, vol. 106, no. 3, pages 282–296, February 2014.

[Maurer 96] T. Maurer & C. von der Malsburg. *Tracking and learning graphs and pose on image sequences of faces*. In International Conference on Automatic Face and Gesture Recognition, October 1996.

[Murphy-Chutorian 07] E. Murphy-Chutorian, A. Doshi & M. Trivedi. *Head Pose Estimation for Driver Assistance Systems: A Robust Algorithm and Experimental Evaluation*. In IEEE Intelligent Transportation Systems Conference, pages 709–714, Sept 2007.

[Murphy-Chutorian 09] E. Murphy-Chutorian & M. M. Trivedi. *Head pose estimation in computer vision: A survey*. IEEE transactions on pattern analysis and machine intelligence, vol. 31, no. 4, pages 607–626, 2009.

[Naik 00] P. Naik & C.-L. Tsai. *Partial least squares estimator for single-index models*. Journal of the Royal Statistical Society B, vol. 62, no. 4, pages 763–771, 2000.

[Oh 05] S. M. Oh, J. M. Rehg, T. Balch & F. Dellaert. *Learning and inference in parametric switching linear dynamic systems*. In IEEE ICCV, 2005.

[Osadchy 07] M. Osadchy, Y. LeCun & M. L. Miller. *Synergistic face detection and pose estimation with energy-based models*. Journal of Machine Learning Research, vol. 8, pages 1197–1215, 2007.

[Pavlovic 00] V. Pavlovic, J. M. Rehg & J. MacCormick. *Learning switching linear models of human motion*. In Conference on Neural Information Processing Systems, 2000.

[Peng 15] X. Peng, J. Huang, Q. Hu, S. Zhang, A. Elgammal & D. Metaxas. *From circle to 3-sphere: Head pose estimation by instance parameterization*. Computer Vision and Image Understanding, vol. 136, pages 92 – 102, 2015.

[Rasmussen 06] C. E. Rasmussen. *Gaussian processes for machine learning*. MIT Press, 2006.

[Raytchev 04] B. Raytchev, I. Yoda & K. Sakaue. *Head pose estimation by nonlinear manifold learning*. In IEEE International Conference on Pattern Recognition, volume 4, pages 462–466, August 2004.

[Salmond 09] D. Salmond. *Mixture reduction algorithms for point and extended object tracking in clutter*. IEEE Transactions on Aerospace and Electronic Systems, vol. 45, no. 2, pages 667–686, 2009.

[Seemann 04] E. Seemann, K. Nickel & R. Stiefelhagen. *Head pose estimation using stereo vision for human-robot interaction*. In IEEE International Conference on Automatic Face and Gesture Recognition, pages 626–631, May 2004.

[Sharma 11] A. Sharma & D. W. Jacobs. *Bypassing synthesis: PLS for face recognition with pose, low-resolution and sketch*. In IEEE Conference onComputer Vision and Pattern Recognition, pages 593–600. IEEE, 2011.

[Smola 04] A. J. Smola & B. Schölkopf. *A tutorial on support vector regression*. Statistics and computing, vol. 14, no. 3, pages 199–222, 2004.

[Srinivasan 02] S. Srinivasan & K. L. Boyer. *Head Pose Estimation Using View Based Eigenspaces*. IEEE International Conference on Pattern Recognition, vol. 4, pages 302–305, 2002.

[Stiefelhagen 04] R. Stiefelhagen. *Estimating head pose with neural networks – results on the pointing04 ICPR workshop evaluation data*. In IEEE International Conference on Pattern Recognition Pointing'04 Workshop, August 2004.

[Sundararajan 15] K. Sundararajan & D. L. Woodard. *Head pose estimation in the wild using approximate view manifolds*. In IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 50–58, June 2015.

[Taheri 13] S. Taheri, A. C. Sankaranarayanan & R. Chellappa. *Joint albedo estimation and pose tracking from video*. IEEE TPAMI, vol. 35, no. 7, pages 1674–1689, July 2013.

[Tipping 99] M. E. Tipping & C. M. Bishop. *Mixtures of probabilistic principal component analyzers*. Neural Computation, vol. 11, no. 2, pages 443–482, February 1999.

[Tu 06] J. Tu, T. Huang & H. Tao. *Accurate head pose tracking in low resolution video*. In IEEE International Conference on Automatic Face and Gesture Recognition, April 2006.

[Uřičář 12] M. Uřičář, V. Franc & V. Hlaváč. *Detector of Facial Landmarks Learned by the Structured Output SVM*. In International Conference on Computer Vision Theory and Applications, February 2012.

[Viola 01] P. Viola & M. Jones. *Rapid object detection using a boosted cascade of simple features*. In IEEE CVPR, June 2001.

[Wang 13] B. Wang, W. Liang, Y. Wang & Y. Liang. *Head Pose Estimation with Combined 2D SIFT and 3D HOG Features*. In International Conference on Image and Graphics, pages 650–655, July 2013.

[Xiong 13] X. Xiong & F. De la Torre. *Supervised descent method and its applications to face alignment*. In IEEE Conference on Computer Vision and Pattern Recognition, pages 532–539, 2013.

[Yang 02] R. Yang & Z. Zhang. *Model-based head pose tracking with stereovision*. In IEEE International Conference on Automatic Face and Gesture Recognition, May 2002.

[Yao 01] P. Yao, G. Evans & A. Calway. *Using affine correspondence to estimate 3-d facial pose*. In IEEE ICIP, October 2001.

[Zhu 12] X. Zhu & D. Ramanan. *Face detection, pose estimation, and landmark localization in the wild*. In IEEE Conference on Computer Vision and Pattern Recognition, pages 2879–2886, June 2012.