



HAL
open science

Projection d'espaces acoustiques : Une approche par apprentissage automatisé de la séparation et de la localisation de sources sonores

Antoine Deleforge

► **To cite this version:**

Antoine Deleforge. Projection d'espaces acoustiques : Une approche par apprentissage automatisé de la séparation et de la localisation de sources sonores. Autre. Université de Grenoble, 2013. Français. NNT : 2013GRENM033 . tel-01134012v2

HAL Id: tel-01134012

<https://theses.hal.science/tel-01134012v2>

Submitted on 21 Mar 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Mathématiques et Informatique**

Arrêté ministériel :

Présentée par

Antoine Deleforge

Thèse dirigée par **Radu Horaud**

préparée au sein de l'Université Joseph Fourier, de l'INRIA Grenoble Rhône-Alpes
et de L'École Doctorale de Mathématiques, Sciences et Technologies de l'Information, Informatique

Acoustic Space Mapping

A Machine Learning Approach to Sound Source Separation and Localization

Thèse soutenue publiquement le **26 Novembre 2013**,
devant le jury composé de :

Pr. Jonathon Chambers

Loughborough University, Rapporteur

Pr. Rémi Gribonval

INRIA Rennes, Rapporteur

Pr. Florence Forbes

INRIA Grenoble Rhône-Alpes, Examinatrice

Pr. Geoff MacLachlan

University of Queensland, Examineur

Pr. Laurent Girin

GIPSA-lab, Grenoble, Examineur

Pr. Radu Horaud

INRIA Grenoble Rhône-Alpes, Directeur de thèse



Acoustic Space Mapping: A Machine Learning Approach to Sound Source Separation and Localization

Projection d'espaces acoustiques: une approche par
apprentissage automatisé de la séparation et de la localisation
de sources sonores

Antoine Deleforge

December 4, 2013

Abstract

In this thesis, we address the long-studied problem of binaural (two microphones) sound source separation and localization through *supervised learning*. To achieve this, we develop a new paradigm referred to as *acoustic space mapping*, at the crossroads of binaural perception, robot hearing, audio signal processing and machine learning. The proposed approach consists in learning a link between auditory cues perceived by the system and the emitting sound source position in another modality of the system, such as the visual space or the motor space. We propose new experimental protocols to automatically gather large training sets that associate such data. Obtained datasets are then used to reveal some fundamental intrinsic properties of acoustic spaces and lead to the development of a general family of probabilistic models for locally-linear high- to low-dimensional space mapping. We show that these models unify several existing regression and dimensionality reduction techniques, while encompassing a large number of new models that generalize previous ones. The properties and inference of these models are thoroughly detailed, and the prominent advantage of proposed methods with respect to state-of-the-art techniques is established on different space mapping applications, beyond the scope of auditory scene analysis. We then show how the proposed methods can be probabilistically extended to tackle the long-known *cocktail party problem*, *i.e.*, accurately localizing one or several sound sources emitting at the same time in a real-word environment, and separate the mixed signals. We show that resulting techniques perform these tasks with an unequalled accuracy. This demonstrates the important role of learning and puts forwards the acoustic space mapping paradigm as a promising tool for robustly addressing the most challenging problems in computational binaural audition.

Résumé

Dans cette thèse, nous abordons les problèmes longtemps étudiés de la séparation et de la localisation binaurale (deux microphones) de sources sonores par l'*apprentissage supervisé*. Dans ce but, nous développons un nouveau paradigme dénommé *projection d'espaces acoustiques*, à la croisée des chemins de la perception binaurale, de l'écoute robotisée, du traitement du signal audio, et de l'apprentissage automatisé. L'approche proposée consiste à apprendre un lien entre les indices auditifs perçus par le système et la position de la source sonore dans une autre modalité du système, comme l'espace visuelle ou l'espace moteur. Nous proposons de nouveaux protocoles expérimentaux permettant d'acquérir automatiquement de grands ensembles d'entraînement qui associent de telles données. Les jeux de données obtenus sont ensuite utilisés pour révéler certaines propriétés intrinsèques des espaces acoustiques, et conduisent au développement d'une famille générale de modèles probabilistes permettant la projection localement linéaire d'un espace de haute dimension vers un espace de basse dimension. Nous montrons que ces modèles unifient plusieurs méthodes de régression et de réduction de dimension existantes, tout en incluant un grand nombre de nouveaux modèles qui généralisent les précédents. Les propriétés et l'inférence de ces modèles sont détaillées en profondeur, et le net avantage des méthodes proposées par rapport à des techniques de l'état de l'art est établi sur différentes applications de projection d'espace, au delà du champ de l'analyse de scènes auditives. Nous montrons ensuite comment les méthodes proposées peuvent être étendues probabilistiquement pour s'attaquer au fameux *problème de la soirée cocktail*, c'est à dire, localiser une ou plusieurs sources sonores émettant simultanément dans un environnement réel, et reséparer les signaux mélangés. Nous montrons que les techniques qui en découlent accomplissent cette tâche avec une précision inégalée. Ceci démontre le rôle important de l'apprentissage et met en avant le paradigme de la projection d'espaces acoustiques comme un outil prometteur pour aborder de façon robuste les problèmes les plus difficiles de l'audition binaurale computationnelle.

ACKNOWLEDGMENT

Je tiens tout d’abord à remercier mon directeur de thèse, Radu Horaud, pour m’avoir fait découvrir la recherche, pour m’avoir appris à rédiger (et non “taper”) de bons articles, pour son soutien constant, son guidage, sa vision, et ses qualités humaines. Je le remercie avant tout pour m’avoir fait confiance, pour avoir cru en mes idées, et pour m’avoir poussé à les mener jusqu’au bout, quitte à prendre des risques. Beaucoup d’éléments de cette thèse n’auraient jamais vu le jour sans son intuition, sa capacité à discerner les bonnes des mauvaises pistes et son soutien scientifique et moral. Je remercie Florence Forbes pour son active collaboration, pour ses connaissances scientifiques et techniques, pour sa patience, ses qualités d’écoute, et de longues discussions intenses, passionnées et fascinantes, parfois jusque tard le soir. Je remercie Laurent Girin pour m’avoir enseigné les bases et en partie transmis la culture du traitement du signal “de la vieille école”, mais aussi pour sa grande ouverture d’esprit et sa forte volonté d’échanger et de collaborer pour aller de l’avant.

I gratefully thank Pr. Rémi Gribonval and Pr. Jonathon Chambers who kindly accepted to review my thesis. I also thank Pr. Geoff MacLachlan who accepted to be part of my thesis committee and to come to my thesis defense all the way from Australia.

I warmly thank Pr. Yoav Schechner for his wonderful welcome at the Technion, for this amazing week-end trip to Golan’s heights on “dust roads” with his adorable kids, for not breaking a tire or running out of fuel in critical moments and for our delightful scientific conversations and “ground breaking” discoveries at the swimming pool. I also thank him a lot for his ability to think “out of the box”, for his guidance, for his crazy but brilliant ideas, and overall, for our very fruitful and efficient collaboration in Haifa. I want to thank Miki Elad for his deep insights on many different subjects, and our very nice scientific discussions and debates. I give a special thank to my friend Yuval Bahat for our very nice collaboration, his smile, his kindness and for hosting me and my friends a couple of times in Tel Aviv. I warmly thank Tamar Galateanu, Orna Nagar-Hillman, Roni Barak and Lee Nudel for making my stay in Israel so smooth and pleasant, as well as my office mates Amit, Yohai, Marina and Vadim for the very enjoyable lunchtime and tight ping-pong games.

Je remercie mes merveilleux parents pour avoir fait de moi ce que je suis, pour leur amour, pour m’avoir relevé dans les moments difficiles, pour avoir cru en moi et pour

m'avoir toujours encouragé à continuer mes études bizarres et incompréhensibles. Je leur dédie humblement cette thèse, en espérant de tout mon coeur que mes recherches déboucheront un jour sur la fabrication d'un sonotone révolutionnaire pour ma maman. Je remercie aussi mes deux petites soeurs que j'aime très fort pour leur soutien, ainsi que leurs pièces rapportées. Je remercie et embrasse ma grand-mère à qui je dois mon goût pour les langues, les voyages, la musique, et bien plus encore. Je remercie enfin mon grand-père qui m'a appris les maths, dont la voix remplie de sagesse ne m'a jamais quitté pendant ces trois années, et m'a guidé mathématiquement et philosophiquement dans les moments les plus obscurs. Je lui dédie toutes les équations de ce manuscrit, car leur existence lui revient sûrement.

Je remercie tous mes collègues de l'INRIA, en commençant par Xavi, son sourire, sa bon humeur, et son caractère fripon. Au delà d'un collègue exemplaire, toujours aidant, pleins d'avis et de conseils enrichissants, c'est un avant tout un très grand ami. Je remercie également Maxime avec qui ce fût un bonheur de partager mon bureau, d'innombrables pauses café, et quelques escapades, sans qui le quotidien de l'INRIA aurait définitivement manqué de galéjades. Un grand merci à Vincent pour avoir su reprendre et comprendre mon code et mes données, et pour en avoir fait de très belles vidéos dont certaines images illustrent cette thèse. Je remercie Soraya Arias pour son expertise et ses conseils survoltés en développement logiciel. Je remercie tout le personnel administratif et particulièrement les différentes assistantes de l'équipe PERCEPTION: Anne Pasteur, Marie-Eve Morency, Florence Polges et Nathalie Gillot, pour leur efficacité à toute épreuve, leur gentillesse et leur patience malgré des ordres de mission toujours plus alambiqués me faisant passer par Tokyo-Turin-Belfast pour une conférence de 3 jours à Chambéry (ou presque). Je remercie Caroline pour le meilleur café du monde et Shakila pour sa bonne humeur et ses délicieux plats. Je remercie toutes les personnes de l'INRIA avec qui j'ai eue le plaisir de collaborer, d'avancer, ou simplement de rire un bon coup, et notamment Jordi pour son humour espiègle, Miles pour son rire inimitable, Israel pour le cheeseboard, Dionysos, Quentin, Pierre, Michel, Ramya, Antoine, Lamia, Guru, Georgios, Ravi, Kaustubh... et tous ceux que j'oublie et auprès de qui je m'excuse.

Pour finir par le plus important, j'exprime un merci colossale à tous mes amis, qui ont constitué les piliers de ma vie pendant ces trois ans, et sans qui je n'aurais pas eue le quart de la force nécessaire pour mener à bien ce travail. En commençant par Raphy et sa grandeur d'âme et d'esprit, qui me fait garder le cap depuis 6 ans et pour qui trois lignes de remerciement ne seront jamais suffisantes. A big thank you to Ahmad and Harsimrat for their craziness and these three lovely years of colocation. Je remercie Laura Adam pour son écoute et son soutien depuis l'autre côté de l'Atlantique. Je remercie Lenny, Maria et Eliya qui m'ont initié à la Bulgarie et au Bulgare, Thomas pour son amitié solide, et Morgane pour la chorale qui guérit. Je remercie Pierre, Julien et Stacy, pour toujours être là quoi qu'il arrive et pour notre amitié intacte depuis 15 ans. Et enfin, un grand merci à Adèle pour ses grands yeux, son sourire, et pour m'avoir supporté et soutenu au quotidien pendant les phases les plus difficiles de la rédaction.

CONTENTS

Acknowledgment	v
List of Acronyms	xi
1 Introduction	1
1.1 Inspiration	1
1.2 Problem Overview and Related Work	3
1.2.1 Localization of sound sources	3
1.2.2 Learning localization through space mapping	5
1.2.3 Separating signals through multiple sound sources localization	6
1.3 Contributions of this thesis	6
1.4 Organization of this Manuscript	8
2 Acoustic Space: Auditory Features, Data and Structure	9
2.1 Definition and Properties	9
2.2 Extraction of Spatial Auditory Features	10
2.2.1 Spectrograms	11
2.2.2 Interaural Spectral Features	11
2.3 Recording Training Data	14
2.3.1 The POPEYE setup	14
2.3.2 Audio-motor acoustic space sampling	15
2.3.3 Audio-visual acoustic space sampling	18
2.4 Manifold Structure of Acoustic Spaces	19
2.4.1 Manifold Learning	20

2.4.2	The duplex theory	21
2.4.3	Audio-motor acoustic space visualization	22
2.4.4	Audio-visual acoustic space visualization	23
2.5	Conclusion	24
3	Probabilistic Space Mapping	25
3.1	Introduction to Space Mapping	25
3.1.1	Regression versus dimensionality reduction	25
3.1.2	Dealing with high-dimensional input	26
3.1.3	Dealing with locally linear data	27
3.1.4	Chapter outline	28
3.2	Gaussian Locally Linear Mapping	28
3.2.1	The GLLiM family of models	28
3.2.2	Link Between GLLiM and Joint Gaussian Mixture Models	29
3.2.3	Forward and inverse mapping functions	30
3.3	Probabilistic Piecewise Affine Mapping	30
3.3.1	Forward versus inverse mapping strategies	30
3.3.2	Geometrical interpretation	32
3.3.3	Expectation-maximization inference for PPAM	33
3.4	Partially Latent Output Mapping: A Hybrid Model	34
3.4.1	Motivation	34
3.4.2	The PLOM model	35
3.4.3	Connection to existing methods	36
3.5	Expectation-maximization inference for PLOM	38
3.5.1	Two data augmentation schemes	38
3.5.2	A note on non-identifiability	39
3.5.3	The general PLOM-EM algorithm	39
3.5.4	The marginal PLOM-EM algorithm	41
3.6	Experiments and Results	43
3.6.1	Evaluation methodology	43
3.6.2	High-dimensional function inversion	44
3.6.3	Robustly retrieving pose and light from face images	47

3.6.4	Retrieval of Mars physical properties from hyperspectral images	50
3.6.5	2D localization of a white noise sound source	54
3.7	Conclusion on Probabilistic Space Mapping	57
3.A	Proof of Theorem 1	59
4	Mapping-Based Sound Source Localization	61
4.1	Sparsity of Natural Sound Spectrograms	62
4.2	Piecewise Constant Mapping	63
4.2.1	Unweighted cost function	63
4.2.2	Normalized cost function	64
4.2.3	Probabilistic Piecewise Constant Mapping	64
4.3	Probabilistic Piecewise Affine Mapping for Spectrograms	66
4.4	Sound Source Localization Results	67
4.4.1	Audio-Motor training set	67
4.4.2	Audio-Visual training set	71
4.4.3	Localization of a human speaker in realistic conditions	72
4.5	Conclusion	74
4.A	Proof of Theorem 2	75
5	Multiple Sound Sources Separation and Localization	77
5.1	Previous Work	77
5.2	Binary Masking	79
5.3	Mixed Probabilistic Piecewise Constant Mapping	81
5.3.1	The mPPCM Model	81
5.3.2	PCESSL: an EM algorithm for mPPCM	81
5.3.3	PCESSL's initialization strategies	83
5.4	Probabilistic Piecewise Affine Inversion in Mixtures	84
5.4.1	The mixed PPAM model	84
5.4.2	The VESSL algorithm	86
5.4.3	VESSL's initialization strategies	87
5.4.4	VESSL's termination	87
5.5	Co-Localization of Sound Source Pairs	88

5.6	Results	89
5.6.1	Tested methods	89
5.6.2	Multiple sound sources separation and localization	89
5.6.3	Co-localization of overlapping source pairs	92
5.7	Conclusion	95
5.A	Detailed Derivations of VESSL	97
6	Conclusion	101
6.1	Summary and Discussion	101
6.2	Direction for Future Research	103
	Publications	105
	International Conference Publications	105
	International Journal Submissions	105
	Other Articles	106
	References	117

LIST OF ACRONYMS

- *1D*: One-dimensional (vector space)
- *2D*: Two-dimensional (vector space)
- *3D*: Three-dimensional (vector space)
- *Avg*: Average (statistics)
- *Az*: Azimuth (angle)
- *CAMIL*: Computational Audio-Motor Integration through Learning (dataset)
- *CoL*: Co-localization (source pair localization algorithm)
- *dB*: Decibel (unit)
- *El*: Elevation (angle)
- *EM*: Expectation-maximization (optimization algorithm)
- *Ex*: Percentage of values higher than a threshold (statistics)
- *E-step*: Expectation step (EM algorithm)
- *GLLiM*: Gaussian Locally Linear Mapping (mapping model)
- *GMM*: Gaussian mixture model (probabilistic model)
- *GPLVM*: Gaussian process latent variable model (mapping model)
- *GTM*: Generative topographic mapping (dimension reduction algorithm)
- *HRTF*: Head Related Transfer Function (acoustic filter)
- *Hz*: Hertz (unit)
- *ICA*: Independant component analysis (source separation algorithm)
- *IEEE*: Institute of electrical and electronics engineers (society)
- *ILD*: Interaural level difference (auditory cue)

- *ILPD*: Concatenated ILD and IPD vectors (auditory cue)
- *IPD*: Interaural phase difference (auditory cue)
- *iPPAM*: Inverse PPAM (SSL algorithm)
- *ISOMAP*: Isometric mapping (dimension reduction algorithm)
- *ITD*: Interaural time delay (auditory cue)
- *ITF*: Interaural transfer function (HRTF ratio)
- *JGMM*: Joint GMM (mapping model)
- *kHz*: kilo-Hertz (unit)
- *kNN*: k-nearest neighbors (graph algorithm)
- *LTSA*: Local tangent space alignment (dimension reduction algorithm)
- *MAP*: Maximum a posteriori (statistics)
- *MESSL*: Model-based EM SSL (SSSL algorithm)
- *MESSL-G*: MESSL with a garbage class (SSSL algorithm)
- *MFA*: Mixture of factor analyzers (dimension reduction algorithm)
- *MLE*: Mixture of local experts (regression algorithm)
- *MLR*: Mixture of linear regressors (regression algorithm)
- *mPPAM*: mixed PPAM (SSSL model)
- *MPPCA*: Mixture of PPCA (dimension reduction algorithm)
- *mPPCM*: mixed PPCM (SSSL model)
- *M-step*: Maximization step (EM algorithm)
- *NRMSE*: Normalized root mean squared error (statistics)
- *Out*: Percentage of values higher than a threshold (statistics)
- *PCA*: Principal component analysis (dimension reduction algorithm)
- *PCCA*: Probabilistic canonical correlation analysis (dimension reduction algorithm)
- *PCESSL*: Piecewise-constant EM SSL (SSSL algorithm)
- *PHAT*: Phase transform (SSL algorithm)
- *PhD*: Philosophiæ doctor (thesis diploma)
- *PLOM*: Partially Latent Output Mapping (mapping model)
- *PLS*: Partial least-squares (regression algorithm)

- *POPEYE*: Perception on purpose eye (robot)
 - *PPAM*: Probabilistic Piecewise Affine Mapping (mapping model)
 - *PPCA*: Probabilistic PCA (dimension reduction algorithm)
 - *PPCM*: Probabilistic piecewise constant mapping (mapping algorithm)
 - *RCA*: Residual component analysis (dimension reduction algorithm)
 - *RGB*: Red-Green-Blue (camera colors)
 - *RVM*: Relevance vector machine (regression algorithm)
 - *SDR*: Signal-to-distortion ratio (source separation score)
 - *SEM*: Stochastic EM (optimization algorithm)
 - *SIR*: Sliced inverse regression (regression algorithm)
 - *SIR*: Signal-to-interferer ratio (source separation score)
 - *SNR*: Signal-to-noise ratio (noise measurement)
 - *SSL*: Sound source localization (task)
 - *SSSL*: Sound source separation and localization (task)
 - *Std*: Standard deviation (statistics)
 - *STFT*: Short-time Fourier transform (spectrogram algorithm)
 - *TDOA*: Time difference of arrival (auditory cue)
 - *TF*: Time-frequency (spectrogram domain)
 - *TIMIT*: Texas instruments and Massachusetts institute of technology (dataset)
 - *TSPD*: Total spectral power density (auditory cue)
 - *VEM*: Variational EM (optimization algorithm)
 - *VESSL*: Variational EM SSL (SSSL algorithm)
 - *WDO*: W-disjoint orthogonality (binary masking assumption)
 - *WN*: White-noise (signal)
-

CHAPTER 1

INTRODUCTION

The biological binaural (two ears) auditory system performs a number of astonishing functions, including spatial immersion, analysis of auditory scenes, precise localization of sound sources or enhancement of desired sources over undesired ones. These functions are of profound interest for technological application and, hence, the subject of increasing engineering efforts. But while human listeners performs these functions daily and effortlessly, reproducing them artificially still constitutes an enigma and fascinating research challenge for computer scientists. Great advances in understanding the biological and neurological properties of the human auditory system allowed to develop efficient computational models inspired from it. But an other important aspect of human hearing is often left apart in the current literature: Human hearing abilities are constantly evolving, readapting, and are the results of years of experience through *learning*. In this thesis, we tackle the challenge of incorporating this process of learning into computational auditory scene analysis. To do so, we propose a novel theoretical and experimental paradigm at the crossroads of binaural perception, robot hearing, audio signal processing and machine learning. This introductory chapter starts by presenting the inspiration and motivation sources of this work. It then provides an overview of the addressed challenges and their associated literature. It finally summarizes the contributions of this thesis and outlines the organization of the remainder of the manuscript.

1.1 Inspiration

This thesis addresses the problem of computational sound source separation and localization using an artificial binaural system. Yet, this work did not start by reading a book on audio signal processing. Nor did it start by browsing articles in machine learning, statistics, robotics or sound source separation. It actually began by naively asking this question: “How do humans localize sounds?”. Specialists of the human auditory system know the answer for more than a century, since a series of observations made by Lord

Rayleigh [Rayleigh 07] where he attempted to account for localization in terms of *inter-aural difference cues*. The listener's head interrupt the sound path from the source to the far ear, resulting in a difference of pressure level and time of arrival (or phase) between the two ears. Latter studies confirmed that specific neurons were dedicated to the measure of such differences[Wang 06]. A simple formula approximately relate the time difference of arrival, the source direction, the speed of sound and the *distance between ears*, and could be used by the brain to localize sounds. At this point, a first oddity is striking: Is the distance between ears hard-coded in our brain? Then, how to account for the fact that this distance is considerably changing from childhood to adulthood? In fact, reality is even more complex. Level and time differences are induced by the complex shape of the head, pinna¹ and torso, and thus vary depending on the emitted sound frequency. Should we therefore consider that an accurate representation of our head shape is neurally encoded? Though, the head's morphology is considerably changing along humans' life, while their ability to localize sounds seems untouched. In fact, some psychological studies [Hofman 98b] showed that even after a strong modification of their outer ear, accurate localization performances were reacquired within days by human subjects. This eliminates the possibility of a static, predefined sound localization pathway in the brain. Despite these evidence, the vast majority of current artificial sound localization methods rely on a fixed geometric or parametric model of sound propagation in the system. Thanks to great advances in the biological and neurological understanding of humans' auditory systems, these models became more and more accurate. Simple models assume a direct path propagation from source to microphones [Yılmaz 04, Liu 10, Alameda-Pineda 12] while more sophisticated ones use a spherical-head model (Woodworth's formula) or a spiral ear model [Kullaib 09]. However, biology teaches us something else: A perfectly functioning auditory system is not enough to understand the acoustic space surrounding us. This is well illustrated by the pathology of *auditory agnosia*. Auditory agnosia manifests primarily in the inability of patients to recognize or differentiate between sounds. It is not a defect of the ear, but a *neurological inability of the brain to process what the sound means*. Much like patients affected with this disability, a raw pair of microphones cannot help much in understanding what and where are the sound sources in real world auditory scenes. In humans, this ability is acquired through *learning* at early stages of development, and is constantly readapted during the life. A nice illustration of this early learning process is the tendency of infants to throw objects around them. Through this simple gesture, the infant associates a voluntary motor action (throwing an object to a specific direction) with a visual sensory input (seeing the place where the object dropped) and an auditory sensory input (hearing the object reaching the floor). In light of this, could learning sound localization be a matter of associating or *mapping* different modalities together?

Things fell into place when I came across an article in psychology [Aytekin 08] published two years before the beginning of my PhD. The central claim of this article may be summarized in a sentence: "A naive organism can *learn* to localize sounds without any *a priori neural representation* of its auditory system, solely based on the *experience*

¹Visible, external part of the ear.

of *voluntary motor actions* on its acoustic inputs”. In other word, the *acoustic space* surrounding a system could be learned by experience, without any knowledge on the system’s parameters. This idea served as a starting point for this work, and motivated the challenge of addressing sound source localization through *learning* using an artificial binaural system. Following this idea will take us to a research journey across the fields of binaural perception, audio signal processing, robot hearing and machine learning. As an outcome, the proposed framework includes a number of new experimental methods, theoretical models, algorithms and techniques that showed promising results for many future developments and applications, and are presented in details in this thesis.

1.2 Problem Overview and Related Work

The human remarkable abilities to localize one or several sound sources and to identify their content from the perceived acoustic signals have been intensively studied in psychophysics [Blauert 97], computational auditory analysis [Wang 06], and more recently in the emerging field of *robot hearing* [Cech 13b]. A classical example that nicely illustrates the difficulty of understanding these human skills, is the well known *cocktail party effect* introduced by Cherry [Cherry 53] that still challenges today’s methods [Haykin 05]: How listeners are able to decipher speech in the presence of other sound sources, including competing talkers? While human listeners solve this problem routinely and effortlessly, this is still a challenge in computational audition. In this thesis, we are interested in a particularly challenging instance of this problem: A binaural system² is placed in a real world, unconstrained environment including reverberations, background noise and competing sources of different types. Two questions are addressed: *what* are the sources and *where* are they located? These tasks are illustrated in Figure 1.1 and may be referred to as the *machine cocktail party problem*. Better addressing this problem is of profound interest for technological application. Typical examples include hearing aids, room acoustics, music technology, robot hearing, and tools for research into auditory physiology and aural perception.

1.2.1 Localization of sound sources

Let us now take a deeper look at the psychophysics of sound localization. There is behavioral and physiological evidence that humans use *interaural* or *binaural cues* in order to estimate the direction of a sound source [Rayleigh 07] and that sound localization plays an important role for solving the cocktail party problem [Middlebrooks 91, Wang 06]. Two binaural cues seem to play an essential role, namely the interaural level difference (ILD) and the interaural time difference (ITD), or its spectral equivalent the interaural phase difference (IPD). Both ILD and IPD are known to be subject-dependent and frequency-dependent cues, due to the so-called *head related transfer function* (HRTF) generated by

²We refer to as *binaural* a system with two microphones that have the particularity of being *embedded* in the device. This notably induces a small distance and possible filtering effects from sound sources to microphones, as opposed to microphones placed far apart in a room without interfering objects.

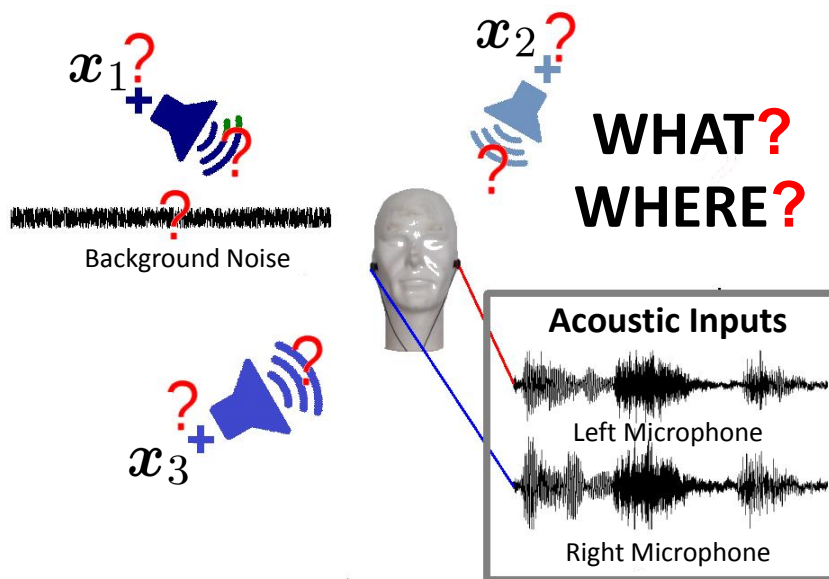


Figure 1.1: The machine cocktail party problem.

the shape of the head, pinna and torso, which filters signals arriving at each eardrum and depending on the sound source direction. It is known that the spatial information provided by interaural-difference cues within a restricted band of frequency is spatially ambiguous, particularly along a roughly vertical and front/back dimension [Middlebrooks 91]. This suggests that humans and mammals make use of full spectrum information for 2D sound source localization [Woodworth 65, Hofman 98a]. This is confirmed by biological models of the auditory system hypothesizing the existence of neurons dedicated to the computation of interaural cues in specific frequency bands [Wang 06].

A lot of computational techniques exist to extract ITD, ILD and IPD from binaural recordings, either in the time domain using cross-correlation [Liu 10, Alameda-Pineda 12], or in the time-frequency domain using Fourier analysis [Mandel 07] or gammatone filters [Woodruff 12]. However, the problem of localizing several sound sources remains a challenge in computational auditory scene analysis, for several reasons. Firstly, the mapping from sound-source positions to interaural cues is usually unknown, complex and non-linear due to the transfer function of microphones which cannot be easily modeled. Secondly, auditory data are corrupted by noise and reverberations. Thirdly, an interaural value at a given frequency is relevant only if the source is actually emitting at that frequency: Natural sounds such as speech are known to be extremely sparse, with often 80% of the frequencies actually missing at a given time. Finally, when several sources emit simultaneously, the assignment of a time-frequency point to one of the sources is not known. The first problem, *i.e.*, mapping audio cues to source positions, is central. Yet, it has received little attention in computational audition. Most existing approaches approximate this mapping based on simplifying assumptions, such as direct-path source-to-microphone propagation [Yilmaz 04], a sine interpolation of ILD data from a human HRTF dataset [Viste 03], or a spiral ear model [Kullaib 09]. These

simplifying assumptions are often not valid in real world conditions. Following this view, accurately modeling a real-world binaural system would require a prohibitively high number of parameters including the exact shape of the recording device, of the room and all their acoustic properties which are not accessible in practice. Due to this difficulty, the vast majority of current binaural sound localization approaches mainly focus on a rough estimation of a frontal azimuth angle, or *one-dimensional* (1D) localization [Liu 10, Mandel 07, Woodruff 12, Viste 03], and very few perform 2D localization [Kullaib 09]. Alternatively, some approaches [Hörnstein 06, Keyrouz 07] bypass the explicit mapping model and perform 2D localization by exhaustive search in a large HRTF look-up table associating source directions to interaural spectral cues. However, this process is unstable and hardly scalable in practice as the number of required associations yields too prohibitive memory and computational costs.

1.2.2 Learning localization through space mapping

A number of psychophysical studies have suggested that the ability of localizing sounds is *learned* at early stages of development in humans and mammals [Hofman 98b, Wright 06, Aytekin 08]. That is, the link between auditory features and source locations would not be hard-coded in the brain but rather learned from experience. One example of such learning processes is the *sensori-motor theory* of perception, originally laid by Poincaré [Poincaré 29] and more recently investigated by O'Regan [O'Regan 01]. This theory suggests that experiencing the sensory consequences of voluntary motor actions is necessary for an organism to learn the perception of space. For example, Held and Hein [Held 63] showed that neo-natal kittens deprived from the ability of moving while seeing could not develop vision properly. Most notably, Aytekin et al. [Aytekin 08] proposed a sensori-motor model of sound source localization using HRTF datasets of bats and humans. In particular, they argue that biological mechanisms could be able to learn sound localization based solely on acoustic inputs and their relation to motor states. Another example of learning process is that of *multimodal fusion*. For instance combining auditory and visual data is naturally performed by human beings. Many behavioral and psychophysical studies [Calvert 04, Ghazanfar 06, Senkowski 08] postulate that the fusion of different sensory modalities is an essential component of perception.

In this thesis, we inspire from these psychological observations and theories to propose a *supervised learning* paradigm for multiple sound source separation and localization. More specifically, we will *train* an artificial system to *map* the space of perceived interaural cues to the space of corresponding source positions, which can be obtained through motor or visual modalities. The task of learning a mapping between two spaces can be summarized as follows: How can we obtain a relationship between two spaces \mathbb{R}^D and \mathbb{R}^L and such that given a new vector observation in \mathbb{R}^D its associated vector in \mathbb{R}^L is deduced? When associated points from the two spaces are given for training, this is referred to as *regression* or *supervised mapping*. When only data from the highest-dimensional space are provided, this is referred to as *dimensionality reduction* or *unsupervised mapping*. In this thesis, a unified formulation of supervised and unsupervised mapping is proposed. A challenging instance of these tasks is when $D \gg L$, *i.e.*, high- to low-dimensional mapping.

This instance is of particular interest in our case since we want to map high-dimensional auditory features to a low-dimensional source position. It has been extensively studied in machine learning, *e.g.*, [Li 91, Xu 95, Tipping 01, Lawrence 05, Rosipal 06], and example of applications are numerous: Motion capture from videos [Agarwal 04, Agarwal 06], sound source localization from acoustic signals [Talmon 11], recovery of physical properties from hyperspectral data [Bernard-Michel 09], to name just a few. A more thorough literature overview on machine learning methods for space mapping is provided in section 3.1.

1.2.3 Separating signals through multiple sound sources localization

An efficient way to compute interaural cues and to map them to a source position is not enough to handle real-world cocktail party scenarios. Indeed, matters are more complicated when different sound sources are simultaneously active, from multiple directions. The sources mix at each microphone and interaural features not only depend on the directions but also on the relative power spectral density of all sources. We therefore need a way to segregate between the different cues, or in other words, to *separate* sound sources. In fact, the problems of sound source localization and separation interleave in a nice way: Localization may help separation by assigning sources to specific interaural cues, and separation may help localization by clustering relevant interaural cues among sources. This dynamic will be extensively used in our work to develop new algorithms allowing to jointly separate and localize multiple emitting sound sources. Note that regardless of the localization problem, sound source separation is of great practical interest on its own, and an immense literature exist on the subject. A overview of this literature and its different aspects is provided in section 5.1.

1.3 Contributions of this thesis

The key novelty of this thesis is to address the long-studied problem of computational binaural sound source separation and localization through learning. Our contributions toward this goal may be decomposed in four main lines

Acoustic Space Mapping We introduce and lay theoretical grounds for the concept of *acoustic space*. The acoustic space of a system is defined as the set of binaural features possibly perceived when sound sources emit in the system's environment. This thesis shows how the intrinsic structure of these spaces as well as their relation to motor or visual modalities can be learned, yielding efficient real-world binaural processing methods. We developed new methodologies to efficiently *sample* acoustic spaces *i.e.*, gather datasets that associate perceived auditory features to sound source positions in other modalities. Some fundamental properties of acoustic spaces are revealed using non-linear dimensionality reduction methods on these datasets. Most notably, we show that they present a smooth locally linear manifold structure parameterized by source positions. This key property suggest that high-dimensional acoustic features could be *mapped*

to low-dimensional sound source positions. This thesis will hence view the sound source localization problem as a space mapping problem. This strongly contrasts with traditional approaches in sound source localization which usually assume the mapping known, based on simplified sound propagation models, *e.g.* [Aarabi 02, Yılmaz 04, Kullaib 09, Liu 10, Alameda-Pineda 12].

Probabilistic Space Mapping through Mixture Models To be applicable to real world sound source localization, a mapping technique should feature a number of properties. First, it should deal with the sparsity of natural sounds, and hence handle missing data. Second, it should deal with the high amount of noise and redundancy present in the interaural spectrograms of natural sounds. Finally, it should allow further extension to the more complex case of mixture of sound sources. An attractive approach embracing all these properties is to use a *Bayesian framework*. In this thesis, we propose a general family of probabilistic model for locally-linear regression, referred to as *Gaussian locally linear mapping* (GLLiM). We show how GLLiM relates to Gaussian mixtures, and thoroughly study several instances of this model. Notably, the mapping may be learned in a supervised way, *i.e.*, associated vectors from both spaces are observed (regression) or in an unsupervised way, *i.e.* only high-dimensional vectors are observed (dimensionality reduction). We propose a new instance of GLLiM that generalizes regression and dimensionality reduction in a unified model referred to as *partially-latent-output mapping* (PLOM). The general inference of PLOM through expectation-maximization procedures is devised. These procedure generalize a number of existing regression and dimensionality reduction techniques. They also provide a large range of new methods that are showed to be advantageous on a wide variety of space mapping tasks, beyond sound source localization. These tasks include synthetic functions inversion, face pose estimation from images and retrieval of Mars' physical properties from hyperspectral data. Our methods are proved to outperform state-of-the-art techniques on these very distinct problems.

Mapping Real-World Acoustic Inputs While space mapping concern vectors, real world acoustic inputs consist in noisy time series of vectors, possibly mixed, and possibly *sparse*, *i.e.*, with a large number of missing values. This missing data problem is handled through adequate probabilistic extensions of the proposed space mapping methods. We also consider a new mapping method referred to as *probabilistic piecewise constant mapping*. To deal with the most complex case of mixture of sound sources, we devise expectation-maximization procedures yielding efficient multiple sound sources separation and localization algorithms. Thorough experiments demonstrate that these algorithms can estimate the two-dimensional direction of one or multiple sound sources emitting simultaneously with an unequaled accuracy in real world conditions. They also outperform several state-of-the art methods in binaural sound source separation.

Beyond Single Source Mapping An experiments conducted towards the end of my PhD yielded surprising results, with potentially strong implications for binaural signal processing in sound mixtures. Pushing further the acoustic space mapping paradigm, we show

that the interaural cues generated by two simultaneous sources could be directly mapped to the directions of *both sources*, without need for separation, even when a very strong time-frequency overlap existed between them. The high localization accuracy obtained with this method opens a new view on the spatial richness of interaural cues.

1.4 Organization of this Manuscript

The remainder of this thesis is organized in four core chapters. Chapter 2 presents in details two experimental protocols to gather acoustic space data. Computational techniques to obtain spatial auditory features from raw binaural signals are showed, and some fundamental properties of acoustic spaces are revealed using manifold learning. Chapter 3 presents the proposed family of probabilistic locally linear space mapping models, as well as general expectation-maximization procedures solving for their inference. Obtained mapping techniques are thoroughly evaluated on a wide range of applications. Chapter 4 presents two mapping-based sound source localization techniques. The first one is based on probabilistic piecewise-constant mapping, while the second one is an extension to spectrogram inputs of the probabilistic piecewise affine mapping technique presented in chapter 3. Experiments show an unequaled accuracy in two-dimensional localization using these methods on real-world scenarios. Chapter 5 devise three extensions of the mapping-based sound source localization methods of chapter 4 to the case of multiple sound sources. Two of these extensions allow for binary-masking based sound source separation. The third extension surprisingly show that recordings of sound source *pairs* can be directly mapped to the two directions of emitting sources, even when a very strong time-frequency overlap exist between emitted signals. Experiments show that proposed methods outperform several state-of-the art techniques both in terms of localization and separation accuracy, using real-world data. These core chapters are followed by a conclusion in Chapter 6, including a summary, discussion, and directions for future works.

An intermediate conclusion on presented contributions and results is provided at the end of each chapter. A list of the international conference publications and international journal submissions made during the thesis is provided at the end of the manuscript. To facilitate the reading, some of the mathematical developments and proofs of theorems are appended at the end of the chapters, after their conclusion. We advise the reader to read this thesis in written order to better appreciate the thought process. However, each chapter may also be read independently and is self-consistent.

CHAPTER 2

ACOUSTIC SPACE: AUDITORY FEATURES, DATA AND STRUCTURE

This chapter introduces and defines the key concept of acoustic space (Section 2.1). We then detail how to extract spatial auditory features from raw binaural signals (Section 2.2). Then, we propose two experimental protocols to gather a large number of such features associated to the position of a sound emitter in the listener’s frame (Section 2.3). This is referred to as *acoustic space sampling*. While these protocols are originally inspired from psychological studies on sound localization learning in humans, they are also designed to be practical and efficient from an engineering point of view. We finally use manifold learning techniques on obtained datasets in order to prove some intrinsic properties of acoustic spaces (Section 2.4). Most notably, we show that they present a locally-linear manifold structure in bijection with the space of source positions. These properties will serve as a basis for the remaining developments in this thesis.

2.1 Definition and Properties

Let us consider a binaural listener, *i.e.*, a pair of audio sensors mounted on a head. Let us assume that these sensors capture auditory feature vectors in \mathbb{R}^D along time. We denote by \mathcal{D} the set of possible directions a sound source can emit from in a listener-centered coordinates frame, *i.e.*, an L -dimensional set of (azimuth,elevation) angle pairs ($L = 2$). Let \mathcal{X} be a closed, connected subset of \mathcal{D} , or any image of such a subset by a smooth bijective function. We define $\mathcal{Y} \subset \mathbb{R}^D$ as the set of auditory features that can be captured by the audio sensors when a single static sound source emits from directions in \mathcal{X} . \mathcal{Y} will be referred to as an *acoustic space* of the binaural system. In this chapter, we experimentally prove that the following properties on \mathcal{Y} are true for some properly chosen auditory features:

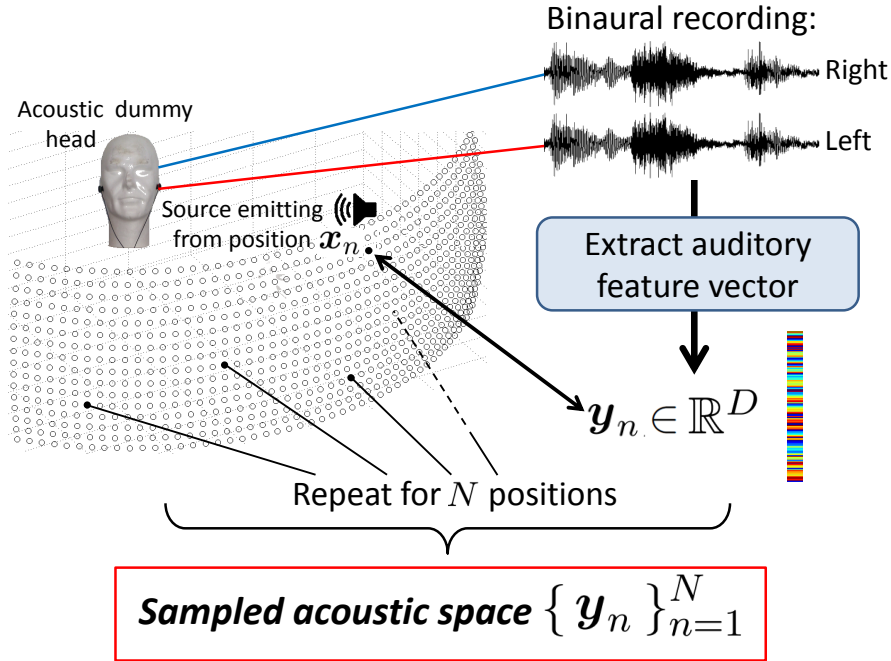


Figure 2.1: Pipeline of acoustic space sampling.

Property 1 \mathcal{Y} is an L -dimensional smooth manifold embedded in the D -dimensional auditory feature space \mathbb{R}^D .

Property 2 There is a smooth, non-linear, but approximately locally-linear bijective mapping g between \mathcal{X} and \mathcal{Y} such that $\mathcal{Y} = g(\mathcal{X})$.

These properties are at the heart of this thesis and will motivate the development of high- to low-dimensional locally-linear space mapping methods in chapter 3, that will be used for sound source localization and separation in chapters 4 and 5.

To thoroughly examine the structure of an acoustic space, we first need to *sample* it, *i.e.*, record a sound source emitting from different directions around the listener, and associate captured auditory features with corresponding directions. An illustration of this is showed in Figure 2.1. Section 2.2 presents the auditory features used, section 2.3 details two different experimental protocols to efficiently sample the acoustic space, and section 2.4 verifies properties 1 and 2 using obtained datasets and a manifold learning technique.

2.2 Extraction of Spatial Auditory Features

An acoustic space can be viewed as a representation of the set of sound source directions around a listener with auditory features. Therefore, these features should be designed to 1) contains as much discriminative spatial information as possible and 2) be as independent of the specific signal emitted as possible. We will refer to such features as *spatial*

auditory features. They can be computed in the time domain, but contain richer information when computed for different frequency channels, *i.e.* in the time-frequency domain. Section 2.2.1 presents the time-frequency representation used in this thesis and section 2.2.2 shows how to compute interaural spectral features using this representation.

2.2.1 Spectrograms

A time-frequency representation can be obtained either using Gammatone filter banks, inspired by human auditory representation as done in *e.g.* [Roman 03, Woodruff 12] or short-term Fourier transform (STFT) analysis, as done in *e.g.* [Wang 05, Mandel 10, Khan 13]. In this thesis we use STFT, notably because it is more directly applicable to sound source separation through binary-masking, as addressed in chapter 5. Spectrograms are computed using a sliding discrete Fourier transform of the raw signal within a specified time window in order to capture the temporal variation of the sound spectrum. They hence discretize signals both in time and frequency.

Three important parameters are to be considered: the sampling frequency, the window length and the window shift. The sampling frequency is the number of sound samples recorded per second. The highest frequency of the spectrogram will be half the sampling frequency. Hence, the sampling frequency governs the spectral range of the spectrogram. The window length is the length of the time window inside which the discrete Fourier transform is computed. The number of positive frequency bins in the spectrogram will be half the number of samples in a window. Hence, the window length governs the frequency resolution. However, using a too large window prevents from capturing *instantaneous* spectral information. The window shift corresponds to the delay between two consecutive windows. The smaller the shift, the higher the resolution of the spectrogram in time, at the cost of a higher computational burden.

In practice the complex-valued spectrograms associated with the two microphones were computed with a 64ms time-window and 8ms window shift, yielding $T = 126$ windows for a 1s signal. Natural sound spectra typically lie in a 0 - 8,000Hz frequency range. Sounds were thus down-sampled to 16,000Hz so that each time window contained 1,024 samples which are transformed into $F = 512$ complex Fourier coefficients associated to positive frequency channels between 0 and 8,000Hz. For a binaural recording made in the presence of a single sound source, we denote with $\{s_{ft}^{(S)}\}_{f,t=1}^{F,T}$ the complex-valued spectrogram emitted by source S, and with $\{s_{ft}^{(L)}\}_{f,t=1}^{F,T}$ and $\{s_{ft}^{(R)}\}_{f,t=1}^{F,T}$ the left and right perceived spectrograms. Figure 2.2(a) shows examples of *total recorded spectral densities* $\{10 \log_{10}(|s_{ft}^{(L)}|^2 + |s_{ft}^{(R)}|^2)\}_{f,t=1}^{F,T}$ for a white-noise (top row) and a speech (bottom row) emitter.

2.2.2 Interaural Spectral Features

Suppose a single sound source S emits from direction $\mathbf{x} \in \mathcal{X}$ in a listener-centered coordinate frame. The *head related transfer function* (HRTF) model provides a relationships

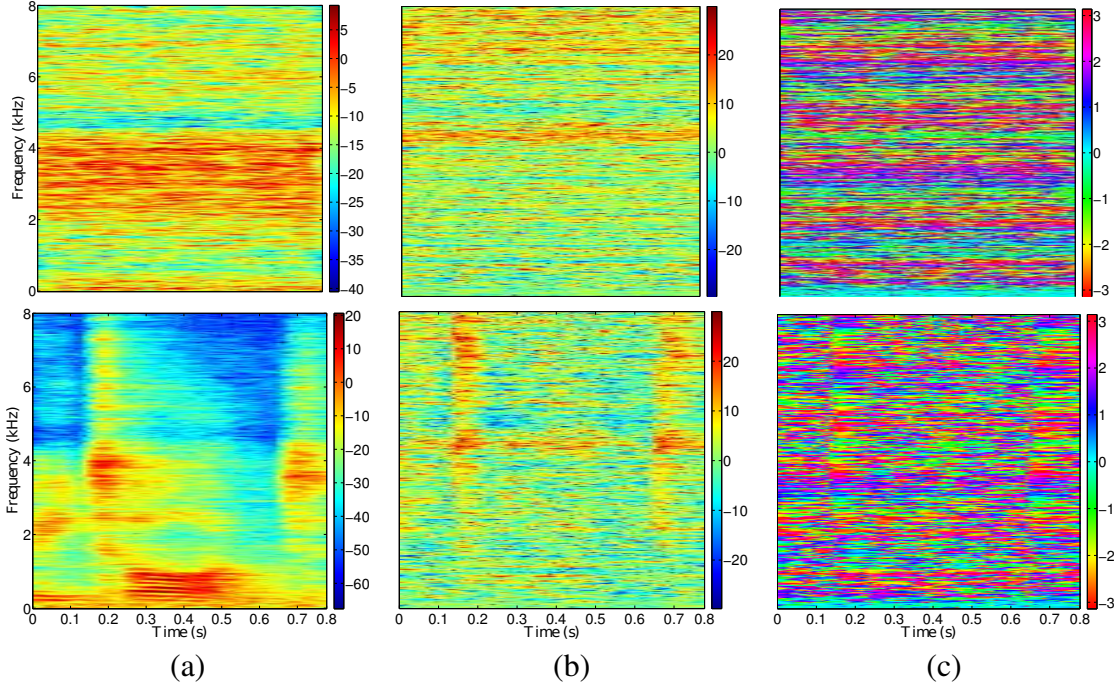


Figure 2.2: Spectrograms obtained from binaural recordings of a single point emitter. First row: White noise emitter, second Row: Speech emitter, first column: Total recorded spectral density, second column: ILD, third column: IPD.

between the spectrogram emitted from source position \mathbf{x} and perceived spectrograms:

$$\begin{cases} s_{ft}^{(L)} = h_f^{(L)}(\mathbf{x}) s_{ft}^{(S)} + e_{ft}^{(L)} \\ s_{ft}^{(R)} = h_f^{(R)}(\mathbf{x}) s_{ft}^{(S)} + e_{ft}^{(R)}. \end{cases} \quad (2.1)$$

where $h^{(L)}$, $h^{(R)}$ denote the left and right non-linear HRTFs, and $e_{ft}^{(L)}$, $e_{ft}^{(R)}$ are terms capturing left and right microphone and background noises. HRTFs are linear acoustic filters with a non-linear dependency with the relative 3D position of the source, due to the complex shapes of the head, pinna, and torso of the listener. However, for sources located in the far field of the listener (> 1.8 meters), [Otani 09] showed that HRTFs mainly depend on the sound source direction while the distance has fewer impact in that case. This is why sound source locations are expressed with 2D angles in this thesis.

The *interaural transfer function* (ITF) is defined by the ratio between the two HRTFs, i.e., $I_f(\mathbf{x}) = h_f^{(R)}(\mathbf{x})/h_f^{(L)}(\mathbf{x}) \in \mathbb{C}$. The interaural spectrogram is defined by $\hat{I}_{ft} = s_{ft}^{(R)}/s_{ft}^{(L)}$. If we assume that noise spectral densities $|e_{ft}^{(L)}|^2$ and $|e_{ft}^{(R)}|^2$ are negligible with respect to the recorded signal, we have

$$\hat{I}_{ft} \approx I_f(\mathbf{x}). \quad (2.2)$$

Under this approximation, \hat{I}_{ft} does not depend on the emitted spectrogram value $s_{ft}^{(S)}$ but only on the emitting source direction \mathbf{x} . We define the *ILD spectrogram* α and the *IPD*

spectrogram ϕ as the log-amplitude and phase of the interaural spectrogram $\hat{I}_{f,t}$:

$$\begin{cases} \alpha_{ft} = 20 \log |\hat{I}_{ft}| \in \mathbb{R}, \\ \phi_{ft} = \arg(\hat{I}_{ft}) \in [-\pi, \pi]. \end{cases} \quad (2.3)$$

Alternatively, we will sometimes express the phase difference in the complex domain, or equivalently \mathbb{R}^2 :

$$\phi'_{ft} = \exp(j \arg(\hat{I}_{ft})) \in \mathbb{C}. \quad (2.4)$$

This expression presents several advantages. It notably allows two nearby phase values to be nearby in terms of Euclidean distance in \mathbb{R}^2 .

Let us now come back to the approximate equality (2.2). It actually holds only if the source is emitting at (f, t) , *i.e.*, $|s_{ft}^{(S)}|^2 > 0$. At low-power TF points, noise spectral densities $|e_{ft}^{(L)}|^2$ and $|e_{ft}^{(R)}|^2$ are dominating. Hence, ILD and IPD do not contain any information about the source position and only capture noise. These points should thus be ignored: we will consider them as *missing values*. They can be determined using, *e.g.*, a threshold on left and right spectral powers $|s_{ft}^{(L)}|^2$ and $|s_{ft}^{(R)}|^2$. Since most *natural* sounds, *e.g.* speech, have a null acoustic level in most time-frequency bins, associated interaural spectrograms have a lot of missing values. Figure 2.2(b,c) depicts the ILD and IPD spectrograms of a single source recording, where the source emits white noise (top row) or speech (bottom row). As can be seen, TF points corresponding to silence in the emitted speech spectrogram yield noisy ILD/IPD values that do not capture spatial information.

In this particular chapter, we put this issue aside and focus on the particular case of a *white-noise* emitter. The theoretical definition of white-noise is a random signal with a flat (constant) power spectral density. In other words, a signal that contains equal power within any frequency channels. However due to the discrete sampling and the final length of spectrogram windows, this is not exactly true in practice. A *practical* white-noise spectrogram takes random values whose modules are positive and temporal means are equal in all frequency channels. Such a signal thus covers the entire acoustic spectrum. We will use white-noise to learn the acoustic space and analyze its properties. It provides data that are more compact and richer, since auditory features are collected in all frequencies at once, rather than, *e.g.*, emitting several consecutive sounds covering different frequency ranges.

Let us consider a static sound source n emitting white noise from $\mathbf{x}_n \in \mathcal{X}$. Since the source is static, interaural features in each time window should be equal due to (2.2). However, some noise is induced by microphones, background noise, and practical white-noise signal properties. To reduce this noise, we can compute the temporal means $\bar{\alpha}_n \in \mathbb{R}^F$ and $\bar{\phi}_n \in \mathbb{R}^{2F}$ of ILD and IPD spectrograms¹. These vectors will be referred to as the *mean interaural feature vectors* associated to \mathbf{x}_n .

¹The temporal mean of IPD values are computed in \mathbb{C} and renormalized to have module 1.

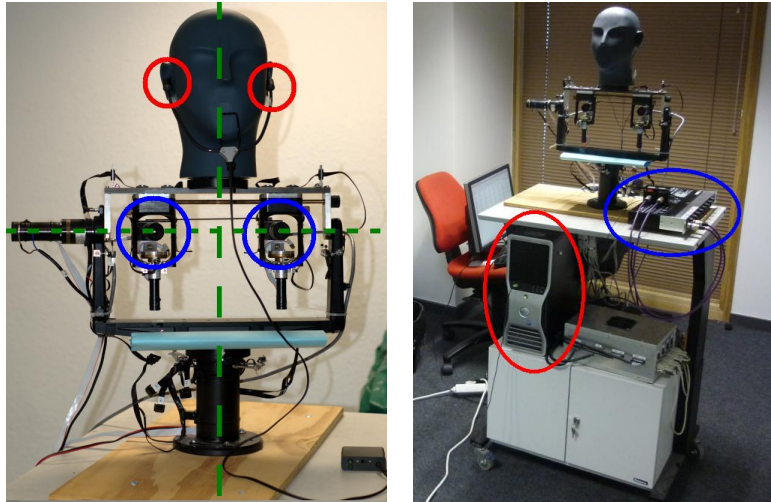


Figure 2.3: Left: The POPEYE setup consists in an acoustic dummy head equipped with a microphone pair (red) and a camera stereo pair (blue). The head is mounted on a motor system with two-degrees of freedom. Pan and tilt motor axis are respectively depicted with a dashed and a dotted green line. Right: Sound acquisition is done via a Behringer ADA8000 Ultragain Pro-8 digital external sound card (blue). Synchronized audio-visual recordings and motor commands are handled by a computer (red).

2.3 Recording Training Data

In this section, we present the recording setup as well as two different methods that we developed to efficiently sample the acoustic space of a listener. The methods were used to record *training data* using a white noise emitter, and *test data* using a natural sound emitter such as speech. The training data will be used to analyse the structure of acoustic spaces in section 2.4 of this chapter and to train the sound source localization and separation algorithms proposed in chapters 4 and 5. The test data, annotated with ground-truth source positions, will be used to evaluate the performance of these algorithms on natural sound recordings.

2.3.1 The POPEYE setup

The POPEYE setup was originally developed within the scope of the European project Perception on Purpose (FP6-IST-027268, January 2006 - December 2008). It consists in an acoustic dummy head mounted on a motor system. The system also includes a stereo camera pair. Pictures of the setup are showed in Figure2.3.

An acoustic dummy head is a device mimicking the shape of a human head, with two microphones in the ear canals. Due to its shape, sounds recorded at microphones are filtered in a similar way as sounds perceived by humans², *i.e.*, with HRTFs (see sec-

²We advice the reader to listen with earphones to acoustic dummy head recordings. It provides a vivid impression to be immersed into a realistic 3-dimensional auditory scene. Such recordings are called *holo-phonic sounds* and can easily be found on the Internet. The most famous one is probably the “barber shop”:

tion 2.2.2). The dummy head used on POPEYE is a Sennheiser MKE 2002 linked to a computer via a Behringer ADA8000 Ultragain Pro-8 digital external sound card.

The head is mounted onto a robotic system with two rotational degrees of freedom. This allows for *pan* motions (left-right, like a “yes”) and *tilt* motions (up-down, like a “no”). The range allowed by motors is $[-180^\circ, +180^\circ]$ in pan angles, and $[-60^\circ, +60^\circ]$ in tilt angles. The system was specifically designed to achieve precise and reproducible movements.

The stereo camera pair mounted on POPEYE allows to achieve stereo reconstruction. However, this will not be useful in the scope of this thesis. Hence, only one of the two camera is used in practice (the left one). The camera captures RGB images with a 480×640 pixels resolution. It has a quite narrow field of view of approximately $21^\circ \times 28^\circ$.

The setup was deliberately placed in a natural, unconstrained room including furniture and echoic walls, *e.g.*, Figure 2.4. This allowed to carry out experiments in real-world conditions, *i.e.*, with natural reverberations and background noise due to, *e.g.*, computer fans.

2.3.2 Audio-motor acoustic space sampling

The first method we developed to gather acoustic space data is referred to as *audio-motor acoustic space sampling*. Resulting datasets are publicly available online under the name *Computational Audio-Motor Integration through Learning (CAMIL)*³. This method was originally inspired by the *sensorimotor contingencies theory* in psychology [O’Regan 01].

<http://www.youtube.com/watch?v=IUDTlvagjJA>.

³http://perception.inrialpes.fr/~Deleforge/CAMIL_Dataset/.

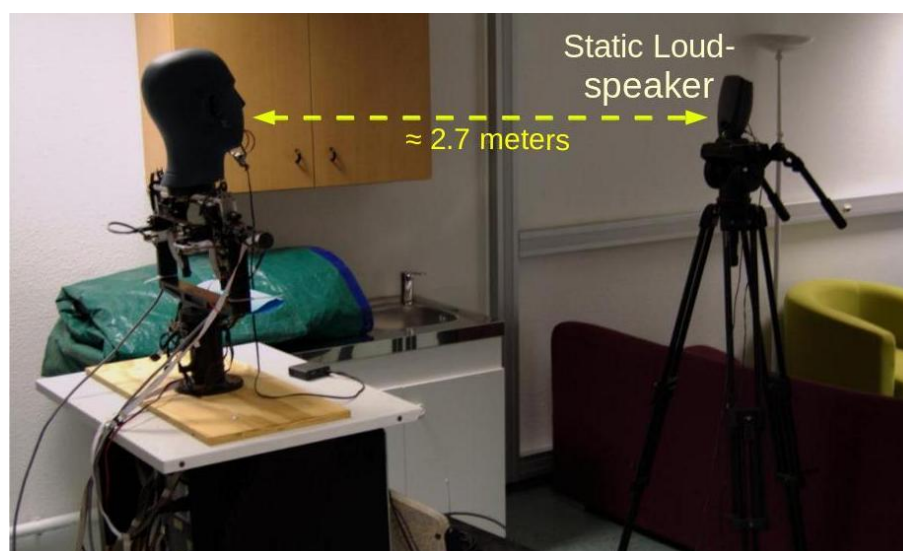


Figure 2.4: Overview of the audio-motor acoustic space sampling setup, in a realistic room environment.

This theory suggests that naive organisms learn perception through experience by associating sensory signals with motor actions (see section 1.2). For instance, as suggested in [Aytekin 08], one would perceive that a sound comes from the left because this sensory event was once associated to the action of turning the head towards the source. A striking illustration of this is a typical behavior of infants: They tend to repeatedly throw around objects. By doing so, they learn an association between the motor action, *i.e.* throwing the object to a specific direction, and a sensory input, *i.e.* hearing the sound of the object on the floor.

Following these ideas, we developed a technique to automatically gather a large number of binaural recordings of a single source associated to the emitter's positions. This is done in an entirely automated way with the POPEYE robot described in previous section. Only the motor system and the microphones are used. The emitter – a loud-speaker – is placed at approximately 2.7 meters ahead of the robot (far field), as showed on Figure 2.4. The loud-speaker's output and the microphones' inputs were handled by two synchronized sound cards in order to simultaneously record and play.

Rather than placing the emitter at known 3D locations around the robot, it was kept in a fixed reference position while the robot recorded emitted sounds *from different motor states*. Consequently, a sound source direction is directly associated to a pan-tilt motor state.

Recordings were made from $N_m = 10,800$ uniformly spread motor states: 180 pan rotations ψ in the range $[-180^\circ, 180^\circ]$ (left-right) and 60 tilt rotations θ in the range $[-60^\circ, 60^\circ]$ (top-down). Hence, the source location spans a 360° azimuth range and a 120° elevation range in the robot's frame, with 2° between neighboring source directions. There is a one-to-one association between motor states and source directions and they will be indifferently denoted by $\{\mathbf{x}_n\}_{n=1}^N \in \mathcal{X}$. Note that here, the space \mathcal{X} has a cylindrical topology. The direct kinematic model of the robot head allows one to easily estimate the 3D position $[p_1; p_2; p_3]$ of the emitter in the robot's frame as a function of pan (β) and tilt (γ) angles:

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} \cos \gamma \cos \beta & -\sin \beta & \cos \beta \sin \gamma \\ \cos \gamma \sin \beta & \cos \beta & \sin \beta \sin \gamma \\ \sin \gamma & 0 & \cos \gamma \end{bmatrix} \begin{bmatrix} d \\ 0 \\ r \end{bmatrix} + r \begin{bmatrix} \cos \beta \sin \gamma \\ \sin \beta \sin \gamma \\ \cos \gamma \end{bmatrix} \quad (2.5)$$

where d is the distance between the microphones' mid point and the emitter when $\gamma = \beta = 0^\circ$, and r is the distance between the microphones' mid point and the tilt axis.

For each $\mathbf{x}_n \in \mathbb{R}^2$, two binaural recordings are made: 1) The loud-speaker emits one second of white noise and 2) the loud-speaker emits a randomly picked utterance amongst 362 samples from the TIMIT dataset [Garofolo 93]. TIMIT utterances are 50% female, 50% male and they last 1 to 5 seconds. Both the head and the loud-speaker are static during the recordings. White noise recordings constitute the *training data* and TIMIT recordings constitute the *test data*.

A great advantage of the audio-motor method is that it allows to obtain a very dense sampling of almost the entire acoustic space. The overall training is fully-automatic and

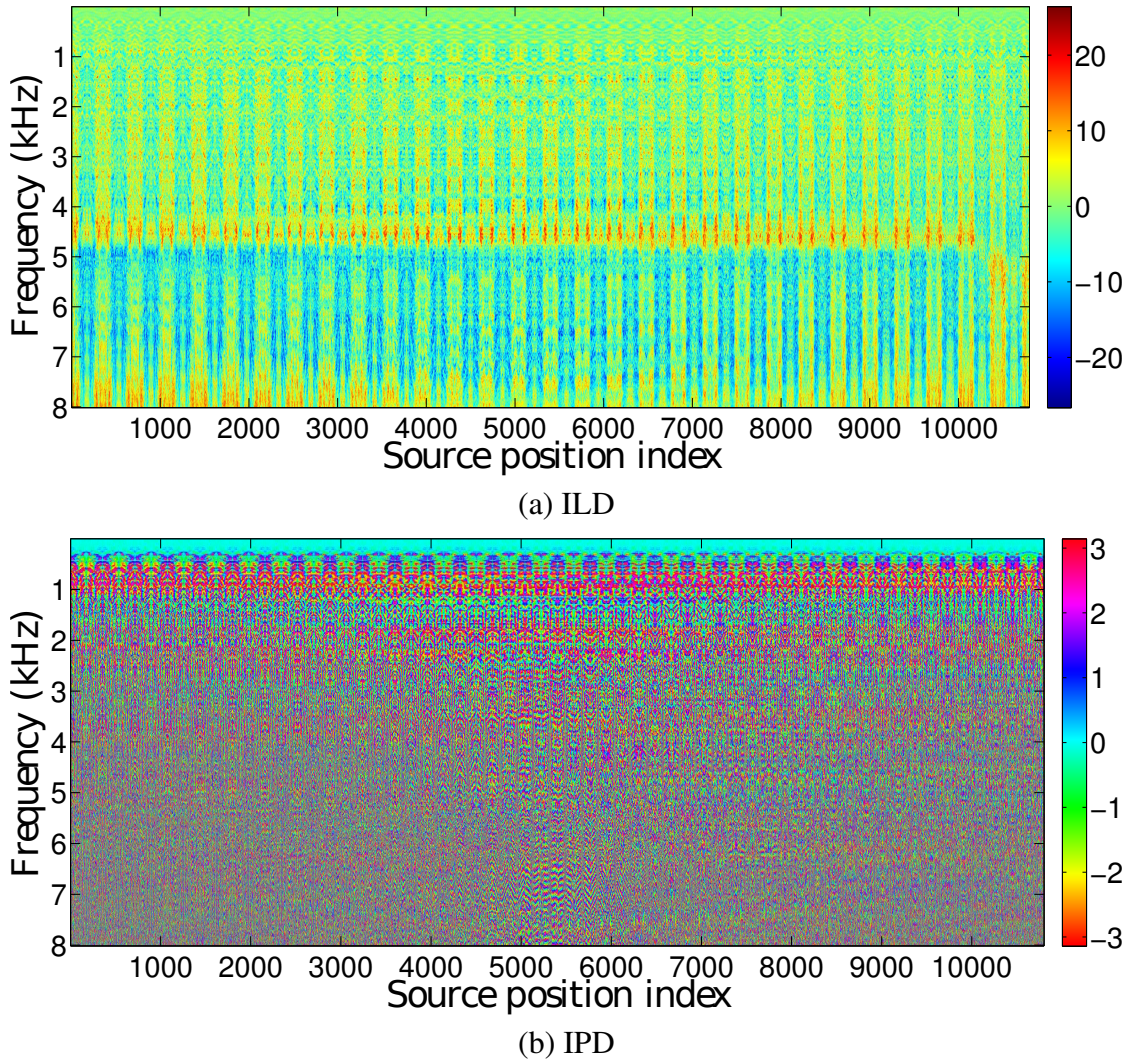


Figure 2.5: Representations of the 10,800 mean ILD (a) and mean IPD (b) feature vectors in the audio-motor dataset. From left-to-right, the source is spanning azimuth values from -180° to $+180^\circ$ at -60° elevation, then its elevation increases of 2° , then the source spans azimuth values from $+180^\circ$ to -180° , its elevation increases, and so on.

takes around 15 hours, which could not be done manually. However, it also presents a limit: a recording made at a given motor-state only approximates what would be perceived if the source was actually moved to the corresponding relative position in the room. This approximation holds only if the room presents relatively few asymmetries and reverberations, which might not be the case in general. Note that when this is the case, a sound localization system trained with this dataset could be used to directly calculate the head movement pointing toward an emitting sound source. This could be done without needing inverse kinematics, distance between microphones or any other parameters.

Figure 2.5 shows the sets of all mean ILD features $\{\bar{\alpha}_n\}_{n=1}^{N_m}$ and mean IPD features $\{\bar{\phi}_n\}_{n=1}^{N_m}$ gathered with the audio-motor space sampling method. A clear and seemingly

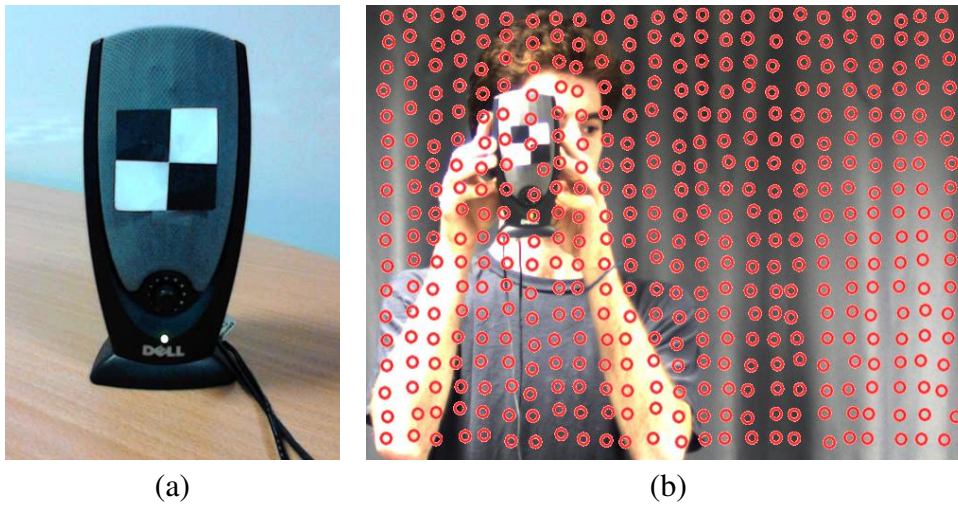


Figure 2.6: (a) Loud-speaker equipped with a chessboard pattern. (b) Image from the audio-visual training sequence. Red circles show the 432 successive positions taken by the loud-speaker.

complex dependency is observed between spatial auditory features and the source position.

2.3.3 Audio-visual acoustic space sampling

The second method used to gather acoustic space data will be referred to as *audio-visual acoustic space sampling*. It is only semi-automatic, but also more realistic than audio-motor training for sound source localization tasks. This time, only the audio-visual part of POPEYE is used without the motors, *i.e.*, the acoustic dummy head and one camera. The setup is used together with an audio-visual source: a loudspeaker fitted with a chessboard pattern that can be easily localized in the camera image. This is showed in Figure 2.6(a).

To obtain training data, the loudspeaker was manually placed at $N_v = 432$ different positions in the camera field-of-view, in a 18×24 grid, as illustrated in Figure 2.6(b). This corresponds to a 23.3 horizontal and vertical pixel spacing between adjacent positions, or equivalently a 1° azimuth and elevation angular spacing. The loud speaker is kept static at each position and emits 1 second of white-noise. Two seconds of silence are kept while moving between each position. The camera and the two microphones are recording all along the experiment. Audio and visual data are synchronized using hand claps at the beginning and at the end of the session. Each 1 second white-noise audio recording can then automatically be cut and associated to the image position of the emitter, by localizing the chessboard pattern in corresponding video frames and averaging its positions. The chessboard pattern is automatically detected using the filter depicted in Figure 2.7. At each pixel intersection (red circle), the intensity of neighboring pixels corresponding to the white part of the filter are subtracted to the intensity of neighboring pixels corresponding to the black part of the filter. The pixel intersection with highest score is then selected as the emitter's position.

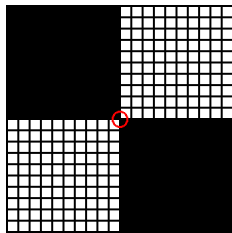


Figure 2.7: Image filter used at each pixel intersection (red circle) to detect the chessboard pattern.

Similarly, we gathered test data by placing the emitter at $9 \times 12 = 108$ positions in the image. At each position, the loudspeaker emitted a 1 to 5 seconds random utterance from the TIMIT dataset [Garofolo 93]. Ground-truth pixel coordinates were obtained thanks to the chessboard pattern.

Contrary to the audio-motor recording approach, the emitter is placed at actual locations in the room, corresponding to 2-dimensional directions in the listener’s frame. Therefore, the audio-visual training set will be suitable for localizing new sounds emitted from the learned region of the room. However, the fact that the emitter must be moved manually restrict the number of positions that can be recorded in practice. For instance, gathering the 432 white-noise recordings took around 25 minutes. A faster but probably less precise approach for training would be to move a continuously emitting white source around the head. Note that the region of the acoustic space that can be sampled is also limited by the camera’s field of view. One could imagine using a camera with a wider field of view to cover a bigger region.

Figure 2.8 shows the sets of all mean ILD features $\{\bar{\alpha}_n\}_{n=1}^{N_m}$ and mean IPD features $\{\bar{\phi}_n\}_{n=1}^{N_m}$ gathered with the audio-visual space sampling method. Again, a clear and seemingly complex dependency is observed between spatial auditory features and the source position.

2.4 Manifold Structure of Acoustic Spaces

While figures 2.5 and 2.8 suggest that some complex dependency exist between spatial auditory features and two-dimensional source positions, they do not reveal the intrinsic structure of these data: Are they linear, non-linear, smooth, discontinuous, discrete?... These questions are hard to answer because we are dealing with very-high dimensional data. However, although interaural feature vectors are high-dimensional, they should be parameterized by 2D source directions. Hence, they should lie on a lower L -dimensional manifold ($L = 2$). We propose to experimentally verify the existence of a Riemannian manifold structure⁴ of acoustic spaces, *i.e.* property 1, by applying manifold learning methods to our data. We then examine whether obtained representations are homeomorphic to the sound source direction space. Such a homeomorphism would allow us to

⁴by definition, a Riemannian manifold is locally homeomorphic to a Euclidean space.

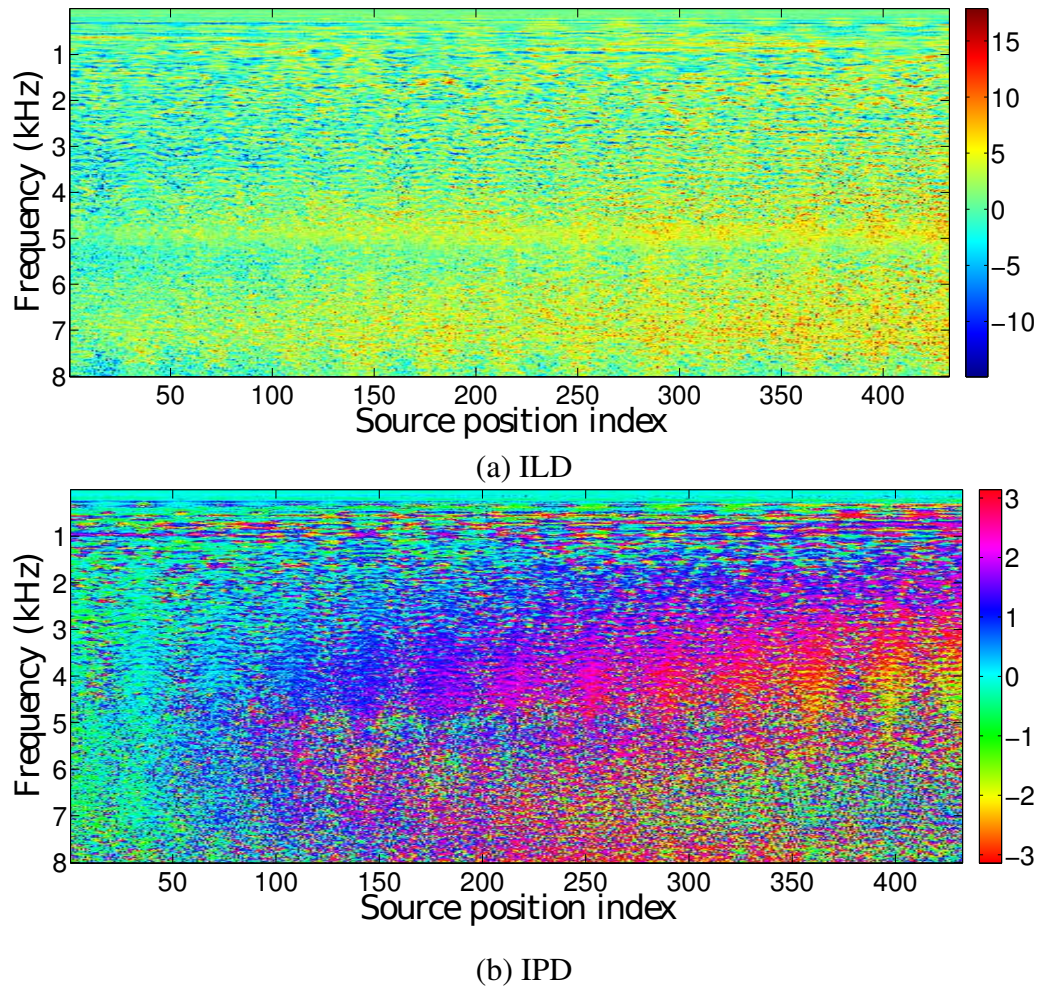


Figure 2.8: Representations of the 432 mean ILD (a) and mean IPD (b) feature vectors in the audio-visual dataset. From left-to-right, the source spans the image from the upper left corner to the lower left corner, then shifts to the left, then spans the image upwards, and so on until the lower-right corner of the image is reached.

confirm (or invalidate) the existence of a locally linear bijective mapping between source directions and the interaural data gathered with our setup, *i.e.*, property 2.

2.4.1 Manifold Learning

If some data lie in a linear low-dimensional subspace of a high-dimensional space, a linear dimensionality reduction method such as principal component analysis (PCA) can be used. In the case of a non-linear subspace, one should use a manifold learning technique, *e.g.*, kernel PCA [Scholkopf 98], ISOMAP [Tenenbaum 00], local-linear embedding [Saul 03], Laplacian eigenmaps [Belkin 03], or local tangent-space alignment (LTSA) [Zhang 04]. We chose to use LTSA because it essentially relies on the assumption that the data are locally linear, which is our central hypothesis. LTSA starts by building a local

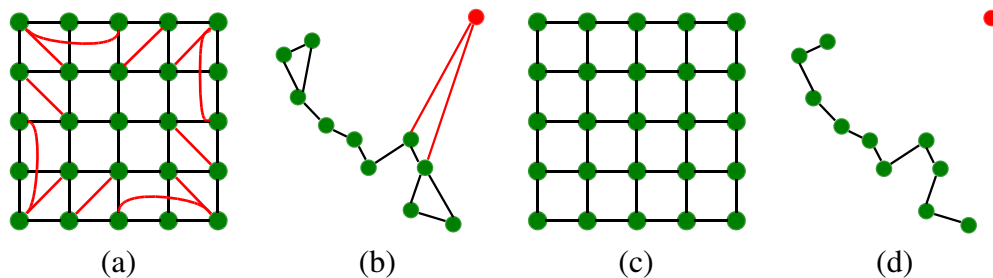


Figure 2.9: Differences between standard k NN (a,b) and symmetric k NN (c,d) on a grid of points with boundaries ($k = 4$) and in the presence of an outlier ($k = 2$).

neighborhood around each high-dimensional observation. If the data lie on a Riemannian manifold, each such neighborhood should span an approximately linear subspace of low dimension L . This corresponds to the tangent space of the manifold around the observation. PCA is then applied to each one of these neighborhoods, yielding as many L -dimensional data representations as points in the data set. Finally a global map is built by optimal alignment of these local representations. This global alignment is done in the L -dimensional space by computing the L largest eigenvalue-eigenvector pairs of a *global alignment matrix* \mathbf{B} (see [Zhang 04] for details).

Two extensions were made to adapt LTSA to our data. First, LTSA uses the *k-nearest neighbours* (kNN) algorithm to determine neighboring relationships between points, yielding neighborhoods of identical size over the data. This has the advantage of always providing connected neighborhood graphs but it can easily lead to inappropriate connections between points, especially at boundaries or in the presence of outliers as showed in Figure 2.9(a) and (b). A simple way to overcome these artifacts is to implement a *symmetric* version of kNN, by considering that two points are connected if and only if each of them belongs to the neighborhood of the other one. Comparisons between the outputs of standard and symmetric kNN are showed in Figure 2.9. Although symmetric kNN solves connexion issues at boundaries, it creates neighborhood of variable sizes, and in particular some points might get disconnected from the graph. Nevertheless, it turns out that detecting such isolated points is an advantage since it may well be viewed as a way to remove outliers from the data. In our case the neighborhood size was set manually.

A second extension was made to represent manifolds which are homeomorphic to the 2D surface of a cylinder. The best way to visualize such a 2D curved surface is to represent it in the 3D Euclidean space and to visualize the 3D points lying on that surface. For this reason, we retained the $L + 1 = 3$ largest eigenvalue-eigenvector pairs of the global alignment matrix \mathbf{B} such that the extracted manifolds can be easily visualized.

2.4.2 The duplex theory

The well established duplex theory [Middlebrooks 91] suggests that ILD cues are mostly used at high frequencies (above 2 kHz) while ITD (or IPD) cues are mostly used at low frequencies (below 2kHz) in humans. Indeed, ILD values are similar at low frequencies

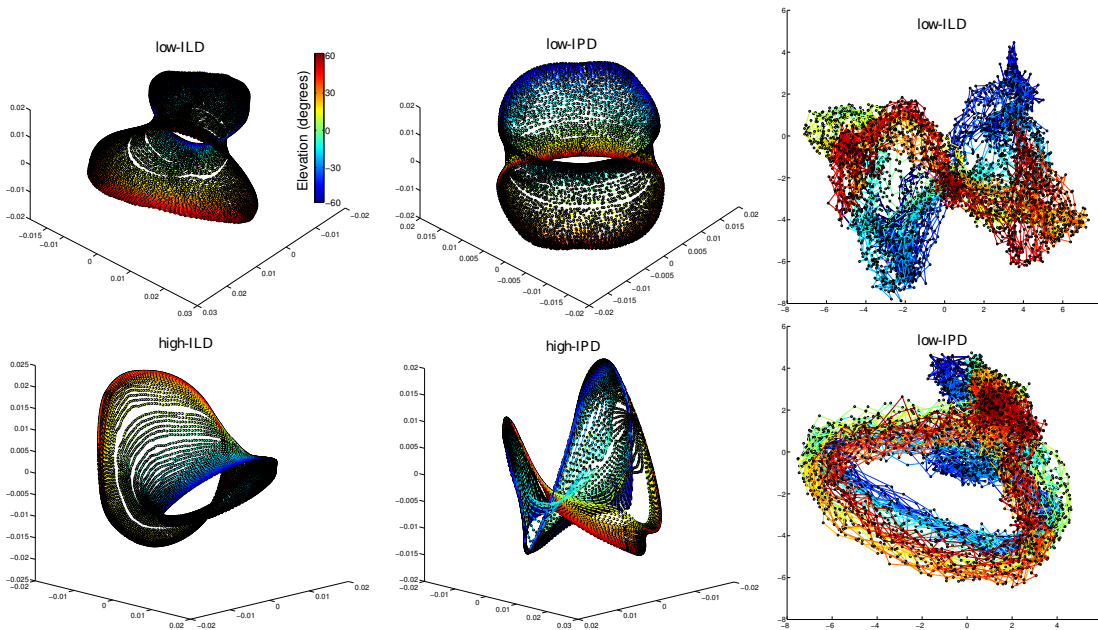


Figure 2.10: 3D representations of mean interaural vectors from the audio-motor dataset using non-linear dimensionality reduction (LTSA). For visualization purpose, points with the same ground truth elevation are linked with a colored line in azimuth order. Obtained point clouds are zero-centered and arbitrarily scaled.

Figure 2.11: 2D representations of frontal mean interaural vectors from the audio-motor dataset using linear dimensionality reduction (PCA) (azimuths in $[-90^\circ, 90^\circ]$).

because the HRTF can be neglected, and the phase difference becomes very unstable with respect to the source position at high frequencies. To account for these phenomena, we chose to analyze low- and high- frequency interaural features separately. The initial interaural features are split into two parts, namely the *low*-ILD and *high*-ILD and the *low*-IPD and *high*-IPD features, where *low* corresponds to frequency channels between 0 and 2kHz and *high* corresponds to frequency channels between 2kHz and 8kHz.

2.4.3 Audio-motor acoustic space visualization

We first applied LTSA to the audio-motor dataset depicted in Figure 2.5. In this dataset, since the source is spanning all possible azimuth directions in $[-180^\circ, +180^\circ]$, the source position space has a cylinder topology. We used neighborhoods of size 20, although any value in the range $[15, 25]$ yielded satisfying results. Maps obtained using LTSA are shown in Figure 2.10. Mean *low*-ILD, *low*-IPD, and *high*-ILD maps are all smooth and homeomorphic to the source direction space (a cylinder), thus confirming properties 1 and 2. However, this is not the case for the mean *high*-IPD map which features more distortions, elevation ambiguities, and crossings. This suggests that high-dimensional IPD data will constitute less reliable cues for sound localization. While the duplex theory is confirmed for IPD cues, the experiments surprisingly show that ILD cues at low frequencies still contain rich enough 2D sound-source position information. This phenomenon was

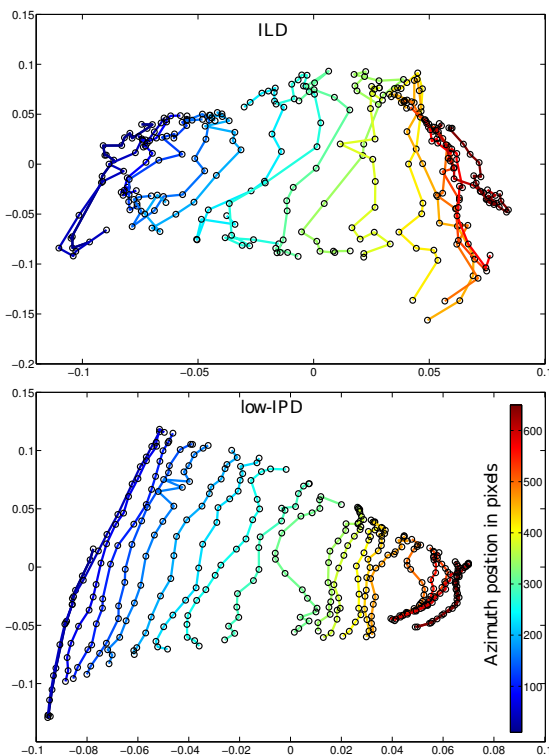


Figure 2.12: 2D representations of mean interaural vectors from the audio-visual dataset using non-linear dimensionality reduction (LTSA). For visualization purpose, points with the same ground truth azimuth are linked with a colored line in elevation order. Obtained point clouds are zero-centered and arbitrarily scaled.

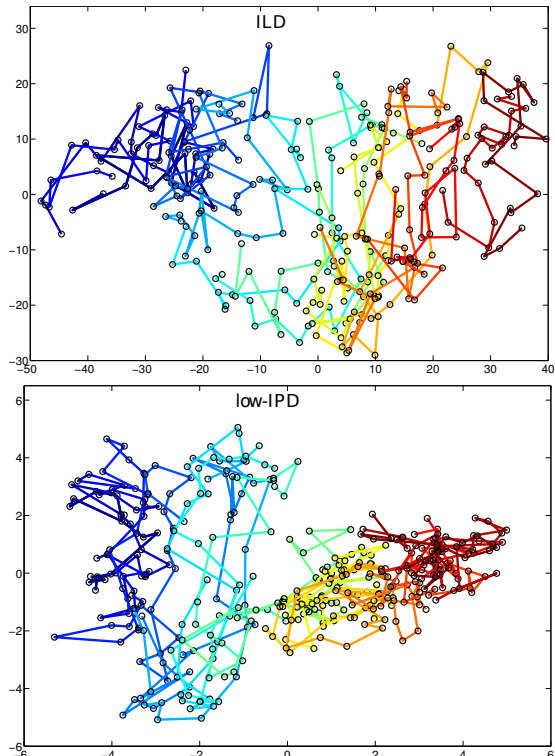


Figure 2.13: 2D representations of mean interaural vectors from the audio-visual dataset using linear dimensionality reduction (PCA).

already noticed in [Mandel 10] but not formally established. These results experimentally prove the existence of *binaural manifolds*, *i.e.*, a strong locally-linear structure hidden behind the complexity of interaural spectral cues obtained from real world recordings in a reverberant room.

For comparison, Figure 2.11 shows the result of applying PCA to mean *low*-ILD vectors and *low*-IPD vectors corresponding to frontal sources (azimuths in $[-90^\circ, 90^\circ]$, 5, 400 points). The resulting maps are extremely distorted, due to the non-linear nature of binaural manifolds. This rules out the use of a linear regression method to estimate the interaural-to-localization mapping and justifies the development of an appropriate piecewise-linear mapping method.

2.4.4 Audio-visual acoustic space visualization

Similarly, we ran LTSA on the audio-visual dataset represented in Figure 2.8. This time, due to the narrow field of view of the camera, the source direction space has a planar

topology. Therefore, we should be able to *unfold* the binaural manifold and represent in 2 dimensions. LTSA was run on the complete-spectrum ILD data (low and high frequency) and on low-IPD data. We used neighborhoods of size 20 for the ILD and 15 for the IPD, because it yielded the best visualizations. Other values in [15, 20] yielded similar maps. Some of the points were manually removed from the audio-visual dataset, because they appeared as outliers using the LTSA visualizations. Maps obtained using LTSA are showed in Figure 2.12. This time, some minor crossings between elevation lines are observed using ILD data. This suggest that ILD features are less discriminative in elevation. Despite this, both ILD and low-IPD maps obtained with LTSA are smooth and homeomorphic to the source position space, *i.e.*, the camera field of view. Comparisons with PCA run on the exact same data are showed in Figure 2.13. Again, the important distortions and crossings in PCA representations suggest that the acoustic space has a non-linear but locally-linear manifold structure.

The audio-motor dataset spans almost the entire acoustic space and captures both changes in sound source position and room position. The audio-visual dataset covers a smaller region and capture changes in sound source position only. Although the two datasets have different properties, the same conclusions are drawn in both case: properties 1 and 2 are experimentally verified using spectral interaural features.

2.5 Conclusion

In this chapter, we introduced the concept of acoustic spaces and presented ways to efficiently sample them. Based on two different datasets, we were able to prove that they possessed two fundamental properties: 1) They have a smooth manifold structure parameterized by source positions and 2) although this manifold is strongly non-linear globally, it is approximately linear locally. This result motivates the development of appropriate locally-linear space mapping methods in order to *learn* the relationship between a high-dimensional and a low-dimensional space. This more general problem is addressed in details in the next chapter. To be applicable to real world sound source localization, the mapping technique should feature a number of properties. First, it should deal with the sparsity of natural sounds, and hence handle missing data. Second, it should deal with the high amount of noise and redundancy present in the interaural spectrograms of natural sounds as opposed to the clean mean interaural vectors obtained from white noise. Finally, it should allow further extension to the more complex case of mixture of sound sources. An attractive approach embracing all these properties is to use a Bayesian framework. This thesis will hence view the sound source localization problem as a probabilistic space mapping problem. This strongly contrasts with traditional approaches in sound source localization which usually assume the mapping known, based on simplified sound propagation models, *e.g.* [Yilmaz 04, Kullaib 09, Liu 10, Alameda-Pineda 12].

CHAPTER 3

PROBABILISTIC SPACE MAPPING

This chapter starts by presenting an overview of space mapping and associated literature in machine learning (Section 3.1). We then introduce a general family of probabilistic model for locally-linear regression, referred to as *Gaussian locally linear mapping* (GLLiM, Section 3.2). We demonstrate a connection between GLLiM and joint Gaussian mixture models, and show that mapping functions in both directions can be obtained from these models. We then justify the advantage of inverse mapping in high-to low-dimensional regression, and subsequently develop in more details a particular instance of GLLiM referred to as *probabilistic piecewise affine mapping* (PPAM) (section 3.3). In section 3.4, a more general model referred to as partially-latent-output mapping (PLOM) is proposed. The key and novel feature of PLOM is that it provides a framework to deal with situations where some of the output’s components can be observed while the remaining components can neither be measured nor be easily annotated. We emphasize that the proposed formulation unifies a number of existing regression and dimensionality reduction methods into a common framework. General EM inference procedures for PLOM are devised in 3.5. We finally compare the proposed PPAM and PLOM methods against state-of-the art regression techniques in section 3.6. The prominent advantage of either PPAM or PLOM is demonstrated on a wide range of problems, including synthetic function inversion, face pose and light estimation from images, retrieval of physical properties from Mars hyperspectral data and white noise sound source localization.

3.1 Introduction to Space Mapping

3.1.1 Regression versus dimensionality reduction

The general task of learning a mapping between a space of input \mathbb{R}^D and a space of output \mathbb{R}^L can be summarized as follows: if we are given training data from one or both spaces,

how can we obtain a relationship between the two such that given a new input observation $\mathbf{y} \in \mathbb{R}^D$, its associated point $\mathbf{x} \in \mathbb{R}^L$ in the other space is deduced? This problem has been extensively studied in machine learning. An interesting and challenging instance is when $D \gg L$, *i.e.*, high- to low-dimensional mapping. Examples of applications are numerous: Motion capture from videos [Agarwal 04, Agarwal 06], sound source localization from acoustic signals [Talmon 11], recovery of physical properties from hyperspectral data [Bernard-Michel 09], to name just a few. We distinguish two types of mapping problems: regression (fully supervised) and dimensionality reduction (fully unsupervised). Regression uses pairs of associated input and output $\{(\mathbf{y}_n, \mathbf{x}_n)\}_{n=1}^N \subset \mathbb{R}^D \times \mathbb{R}^L$ as training data, and the task is to infer a relationship $\mathbf{x} = \mathbf{g}(\mathbf{y})$. In dimensionality reduction, only high-dimensional data $\{\mathbf{y}_n\}_{n=1}^N \subset \mathbb{R}^D$ are used for training, and an associated latent low-dimensional representation $\{\mathbf{x}_n\}_{n=1}^N \subset \mathbb{R}^L$ is sought. Although regression and dimensionality reduction are usually treated distinctly, we propose a unified formulation in this chapter and refer to them as *supervised* and *unsupervised space mapping*. This unified formulation leads to a new hybrid model where the output is *partially latent*. High- to low-dimensional regression methods emerging from this formulation are thoroughly detailed, derived and evaluated in this chapter.

Both supervised and unsupervised mapping have been addressed using a probabilistic model, *i.e.*, \mathbf{y} and \mathbf{x} are modeled as realizations of random variables \mathbf{Y} and \mathbf{X} . This approach is chosen because it presents a number of advantages for the sound source separation and localization problems addressed in this thesis. Most notably, a probabilistic model straightforwardly deals with missing data, can be easily combined with other models, and a mixture of models can be considered. This will be necessary to deal with non-vector input such as spectrograms in chapter 4, and to address the more complex case of sound source mixtures in chapter 5. Nevertheless, this chapter aims at developing general high-to-low dimensional regression methods for vector-valued data. The proposed methods will be tested in section 5.6 on various datasets and applications, without particular emphasis on audio signal processing.

3.1.2 Dealing with high-dimensional input

Estimating a function having a high-dimensional support is generally hard, because for most regression methods, *e.g.* polynomial interpolation, this will imply the estimation of a huge number of parameters. This is why existing methods often solve for high-to-low dimensional regression in two steps: dimension reduction followed by regression. This presents a risk to map the input \mathbf{Y} onto an intermediate low-dimensional space that does not necessarily contain the information needed to correctly predict the output \mathbf{X} . To prevent this risk, a number of methods perform the dimension reduction step by taking the output variable into account. The concept of *sufficient reduction* [Cook 07] was specifically introduced for solving regression problems. The action of replacing the input with a lower-dimensional representation is called *sufficient dimension reduction* when this action retains all relevant information about the output. Methods falling into this category are partial least-squares (PLS) [Rosipal 06], sliced inverse regression (SIR) [Li 91], kernel SIR [Wu 08], and principal component based methods [Cook 07, Adragni 09]. SIR

methods are not designed specifically for prediction and do not provide a specific predictive method. Once a dimension reduction has been determined, any standard method can be used to perform predictions, which is likely to be sub-optimal as it is not necessarily consistent with the reduction model. Regarding PLS, its superior performance over standard principal component regression is subject to the relationship between the covariance of \mathbf{X} and \mathbf{Y} , and the eigen-structure of the covariance of \mathbf{Y} [Naik 00]. The principal component methods proposed in [Cook 07, Adraghi 09] are based on a semi-parametric model of $\mathbf{Y}|\mathbf{X} = \mathbf{x}$ and can be used without specifying a model for the joint distribution of \mathbf{X} and \mathbf{Y} . In general, we believe that a direct approach is always more preferable than a two-step approach. Indeed the latter adds complexity and cannot be interpreted in terms of a single optimization problem.

In this chapter, we come up with a single-step approach by taking the problem at hand the other way round: We probabilistically model the high-dimensional input as a smooth function of low-dimensional output, *i.e.* we assume that high-dimensional data lie on a low-dimensional manifold. Since in that direction the function has a low dimensional support, the model requires a relatively small number of parameters, *i.e.*, linear in D . We show that a learned optimal model can then be used to obtain the inverse function through Bayes' inversion, thus leading to the desired high- to low-dimensional regression.

3.1.3 Dealing with locally linear data

To deal with non-linear data, a common approach in the literature is to use kernel methods. These methods represent observed data in a high-dimensional, possibly infinite-dimensional feature space. This is done by defining a *kernel function* over the observed space that replaces the dot product. Since the kernel function can be quite general and not necessarily linear, the relations found in this way are accordingly very general. Some examples of kernel methods for regression are kernel SIR [Wu 08], the relevance vector machine method [Tipping 01] or its multivariate extension [Thayananthan 06]. Amongst kernel methods, Gaussian process latent variable models (GPLVM) form a widely used family of probabilistic mapping models. GPLVM was originally formulated in [Lawrence 05] as a dimensionality reduction method. It can be viewed as a non-linear probabilistic version of principal component analysis (PCA). It was later extended to deal with regression problems [Fusi 12, Wang 12]. A drawback of kernel methods is that they require the choice of an appropriate kernel function, which cannot be done automatically and highly depend on the data considered. Moreover, the non-linearity of kernel functions limit the number of possible extensions. For example, mappings learned with the methods listed above cannot be inverted.

An other attractive approach for modeling non-linear mapping problems probabilistically is to use a mixture of *locally linear* models. In the Gaussian case, we show in this chapter that it boils down to estimating a Gaussian mixture model (GMM) on the joint input and output spaces. We will refer to the corresponding family of mapping models as *supervised Gaussian locally linear mapping* (GLLiM) in the case of regression and *unsupervised GLLiM* in the case of dimensionality reduction. We show in section 3.4.3 of

this chapter that a number of existing regression [de Veaux 89, Kain 98, Qiao 09] and dimensionality reduction [Tipping 99b, Tipping 99a, Ghahramani 96, Bach 05, Bishop 98, Kalaitzis 12] methods may be viewed as particular instances of GLLiM.

3.1.4 Chapter outline

In section 3.2 we present in detailed the GLLiM family of models. We show that two strategies are available when using them for space mapping, namely the *forward mapping* and the *inverse mapping* strategy. In section 3.3, we show that although the inverse mapping strategy has been overlooked in the literature, it is particularly natural and advantageous in the case of high- to low-dimensional regression. This results in a new regression method that will be referred to as *probabilistic piecewise affine mapping* (PPAM). An *expectation-maximization* (EM) algorithm for the inference of PPAM is devised in section 3.3.3. In section 3.4, we propose a new model referred to as *partially-latent-output mapping* (PLOM). This model generalizes supervised and unsupervised GLLiM. We show that PLOM encompasses a large number of regression and dimensionality reduction models, including PPAM. It also provides a range of new hybrid models allowing to deal with practical regression problems where the output can only be partially observed. General EM inference methods solving for PLOM are studied and devised in section 3.5. Section 3.6 tests and compares PPAM and PLOM with a number of state-of-the-art regression techniques via experiments performed on synthetic data, on a dataset of 3D faces, on hyper-spectral images of the Mars surface and on interaural spectral features. Various experimental conditions showing the advantage of PPAM or PLOM are studied. Section 3.7 concludes this chapter on probabilistic space mapping.

3.2 Gaussian Locally Linear Mapping

3.2.1 The GLLiM family of models

Let $\mathcal{X} \subseteq \mathbb{R}^L$ denotes a low-dimensional space and \mathbb{R}^D a high-dimensional space. Suppose there exists a smooth, locally linear bijection $\mathbf{g} : \mathcal{X} \rightarrow \mathcal{Y} \subset \mathbb{R}^D$ such that the set $\mathcal{Y} = \{\mathbf{g}(\mathbf{x}), \mathbf{x} \in \mathcal{X}\}$ forms an L -dimensional manifold embedded in \mathbb{R}^D . GLLiM methods rely on a piecewise linear assumption, *i.e.*, any realization (\mathbf{y}, \mathbf{x}) of $(\mathbf{Y}, \mathbf{X}) \in \mathbb{R}^D \times \mathbb{R}^L$ is such that \mathbf{y} is the image of $\mathbf{x} \in \mathcal{R}_k$ by an affine transformation τ_k , plus an error term. If we assume that there is a finite number K of affine transformations τ_k and an equal number of associated local regions $\mathcal{R}_k \subset \mathbb{R}^L$, we obtained a piecewise-affine approximation of \mathbf{g} . This is modeled by a hidden variable Z such that $Z = k$ if and only if \mathbf{Y} is the image of $\mathbf{X} \in \mathcal{R}_k$ by τ_k . If \mathbb{I} is the indicator function such that $\mathbb{I}(Z = k) = 1$ if $Z = k$ and 0 otherwise, it follows that:

$$\mathbf{Y} = \sum_{k=1}^K \mathbb{I}(Z = k)(\mathbf{A}_k \mathbf{X} + \mathbf{b}_k + \mathbf{E}_k) \quad (3.1)$$

where matrix $\mathbf{A}_k \in \mathbb{R}^{D \times L}$ and vector $\mathbf{b}_k \in \mathbb{R}^D$ define the parameters of the affine transformation τ_k and $\mathbf{E}_k \in \mathbb{R}^D$ is an error term capturing both the observation noise in \mathbb{R}^D and the reconstruction error due to the local affine approximation. Under the assumption that \mathbf{E}_k is a zero-mean Gaussian with covariance matrix $\Sigma_k \in \mathbb{R}^{D \times D}$ and that it does not depend on $\mathbf{X}, \mathbf{Y}, Z$, we obtain:

$$p(\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x}, Z = k; \boldsymbol{\theta}) = \mathcal{N}(\mathbf{y}; \mathbf{A}_k \mathbf{x} + \mathbf{b}_k, \Sigma_k) \quad (3.2)$$

where $\boldsymbol{\theta}$ designates the vector of model parameters. To complete the hierarchical definition of the joint distribution $p(\mathbf{Y}, \mathbf{X}, Z; \boldsymbol{\theta})$ and make the affine transformations local, the regions $\{\mathcal{R}_k\}_{k=1}^K$ are modeled in a probabilistic way by assuming that \mathbf{X}_n follows a mixture of K Gaussians defined by

$$p(\mathbf{X} = \mathbf{x} | Z = k; \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}; \mathbf{c}_k, \Gamma_k) \quad \text{and} \quad p(Z = k; \boldsymbol{\theta}) = \pi_k \quad (3.3)$$

with $\mathbf{c}_k \in \mathbb{R}^L$, $\Gamma_k \in \mathbb{R}^{L \times L}$, and $\sum_{k=1}^K \pi_k = 1$. The set of GLLiM's parameters is:

$$\boldsymbol{\theta} = \{\mathbf{c}_k, \Gamma_k, \pi_k, \mathbf{A}_k, \mathbf{b}_k, \Sigma_k\}_{k=1}^K. \quad (3.4)$$

3.2.2 Link Between GLLiM and Joint Gaussian Mixture Models

When parameters $\boldsymbol{\theta}$ are unconstrained, we can prove that the joint distribution $p(\mathbf{X}, \mathbf{Y}; \boldsymbol{\theta})$ defined by (3.2) and (3.3) is an unconstrained Gaussian mixture model (GMM) on the joint variable $[\mathbf{X}; \mathbf{Y}]$ ($[\cdot; \cdot]$ denotes vertical concatenation). This model is referred to as the joint GMM (JGMM) model in the acoustic and speech domains, *e.g.*, [Kain 98, Qiao 09]. This statement is formalized by the following theorem:

Theorem 1 *A GLLiM model on \mathbf{X}, \mathbf{Y} with unconstrained parameters $\boldsymbol{\theta}$ is equivalent to a Gaussian mixture model on the joint variable $[\mathbf{X}; \mathbf{Y}]$ with unconstrained parameters $\boldsymbol{\psi} = \{\mathbf{m}_k, \mathbf{V}_k, \rho_k\}_{k=1}^K$, *i.e.*,*

$$p(\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}; \boldsymbol{\theta}) = \sum_{k=1}^K \rho_k \mathcal{N}([\mathbf{x}; \mathbf{y}]; \mathbf{m}_k, \mathbf{V}_k). \quad (3.5)$$

The parameters $\boldsymbol{\theta}$ can be expressed as a function of $\boldsymbol{\psi}$ by:

$$\begin{aligned} \pi_k &= \rho_k, \quad \mathbf{c}_k = \mathbf{m}_k^x, \quad \Gamma_k = \mathbf{V}_k^{xx}, \quad \mathbf{A}_k = \mathbf{V}_k^{xy\top} \mathbf{V}_k^{xx-1}, \quad \mathbf{b}_k = \mathbf{m}_k^y - \mathbf{V}_k^{xy\top} \mathbf{V}_k^{xx-1} \mathbf{m}_k^x, \\ \Sigma_k &= \mathbf{V}_k^{yy} - \mathbf{V}_k^{xy\top} \mathbf{V}_k^{xx-1} \mathbf{V}_k^{xy}, \quad \text{where } \mathbf{m}_k = \begin{bmatrix} \mathbf{m}_k^x \\ \mathbf{m}_k^y \end{bmatrix} \quad \text{and} \quad \mathbf{V}_k = \begin{bmatrix} \mathbf{V}_k^{xx} & \mathbf{V}_k^{xy} \\ \mathbf{V}_k^{xy\top} & \mathbf{V}_k^{yy} \end{bmatrix}. \end{aligned} \quad (3.6)$$

The parameters $\boldsymbol{\psi}$ can be expressed as a function of $\boldsymbol{\theta}$ by:

$$\rho_k = \pi_k, \quad \mathbf{m}_k = \begin{bmatrix} \mathbf{c}_k \\ \mathbf{A}_k \mathbf{c}_k + \mathbf{b}_k \end{bmatrix} \quad \text{and} \quad \mathbf{V}_k = \begin{bmatrix} \Gamma_k & \Gamma_k \mathbf{A}_k^\top \\ \mathbf{A}_k \Gamma_k & \Sigma_k + \mathbf{A}_k \Gamma_k \mathbf{A}_k^\top \end{bmatrix}. \quad (3.7)$$

A proof of Theorem 1 is given in appendix 3.A.

3.2.3 Forward and inverse mapping functions

Given θ , a mapping from \mathbb{R}^L to \mathbb{R}^D is obtained using the *forward conditional density*, i.e.,

$$p(\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x}; \theta) = \sum_{k=1}^K \frac{\pi_k \mathcal{N}(\mathbf{x}; \mathbf{c}_k, \mathbf{\Gamma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}; \mathbf{c}_j, \mathbf{\Gamma}_j)} \mathcal{N}(\mathbf{y}; \mathbf{A}_k \mathbf{x} + \mathbf{b}_k, \mathbf{\Sigma}_k) \quad (3.8)$$

while a mapping from \mathbb{R}^D to \mathbb{R}^L is obtained using the *inverse conditional density*, i.e.,

$$p(\mathbf{X} = \mathbf{x} | \mathbf{Y} = \mathbf{y}; \theta) = \sum_{k=1}^K \frac{\pi_k \mathcal{N}(\mathbf{y}; \mathbf{c}_k^*, \mathbf{\Gamma}_k^*)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{y}; \mathbf{c}_j^*, \mathbf{\Gamma}_j^*)} \mathcal{N}(\mathbf{x}; \mathbf{A}_k^* \mathbf{y} + \mathbf{b}_k^*, \mathbf{\Sigma}_k^*), \quad (3.9)$$

where:

$$\begin{aligned} \mathbf{c}_k^* &= \mathbf{A}_k \mathbf{c}_k + \mathbf{b}_k, \quad \mathbf{\Gamma}_k^* = \mathbf{\Sigma}_k + \mathbf{A}_k \mathbf{\Gamma}_k \mathbf{A}_k^\top, \quad \mathbf{A}_k^* = \mathbf{\Sigma}_k^* \mathbf{A}_k^\top \mathbf{\Sigma}_k^{-1}, \\ \mathbf{b}_k^* &= \mathbf{\Sigma}_k^* (\mathbf{\Gamma}_k^{-1} \mathbf{c}_k - \mathbf{A}_k^\top \mathbf{\Sigma}_k^{-1} \mathbf{b}_k) \quad \text{and} \quad \mathbf{\Sigma}_k^* = (\mathbf{\Gamma}_k^{-1} + \mathbf{A}_k^\top \mathbf{\Sigma}_k^{-1} \mathbf{A}_k)^{-1}. \end{aligned} \quad (3.10)$$

Note that given an observation in one space, both (3.8) and (3.9) take the form of a Gaussian mixture distribution in the other space. These Gaussian mixtures are parameterized in two different ways by the observed data and the GLLiM parameter vector θ . One can use their expectation to obtain *forward* and *inverse* mapping functions:

$$\mathbb{E}[\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x}; \theta] = \sum_{k=1}^K \frac{\pi_k \mathcal{N}(\mathbf{x}; \mathbf{c}_k, \mathbf{\Gamma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}; \mathbf{c}_j, \mathbf{\Gamma}_j)} (\mathbf{A}_k \mathbf{x} + \mathbf{b}_k) \quad (3.11)$$

$$\mathbb{E}[\mathbf{X} = \mathbf{x} | \mathbf{Y} = \mathbf{y}; \theta] = \sum_{k=1}^K \frac{\pi_k \mathcal{N}(\mathbf{y}; \mathbf{c}_k^*, \mathbf{\Gamma}_k^*)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{y}; \mathbf{c}_j^*, \mathbf{\Gamma}_j^*)} (\mathbf{A}_k^* \mathbf{y} + \mathbf{b}_k^*) \quad (3.12)$$

3.3 Probabilistic Piecewise Affine Mapping

3.3.1 Forward versus inverse mapping strategies

Let us now focus on the situation we are interested in, i.e., high- to low-dimensional regression from \mathbb{R}^D to \mathbb{R}^L ($D \gg L$). Given a training set of observed input-output pairs $\{(\mathbf{y}_n, \mathbf{x}_n)\}_{n=1}^N \subset \mathbb{R}^D \times \mathbb{R}^L$, we suppose that a set of parameters maximizing the GLLiM observed-data log-likelihood $\mathcal{L}(\theta) = \log p(\{\mathbf{y}_n, \mathbf{x}_n\}_{n=1}^N; \theta)$ can be learned. Since both forward and inverse mapping functions are available, it is natural to consider two strategies:

1. Learn parameters $\tilde{\theta}$ corresponding to a mapping from \mathbf{X} to \mathbf{Y} using the model presented in section 3.2.1, i.e., \mathbf{Y} is a piecewise-affine transformation of \mathbf{X} . Then use the inverse mapping function (3.12).

2. Learn parameters $\tilde{\boldsymbol{\theta}}^*$ corresponding to a mapping from \mathbf{Y} to \mathbf{X} , *i.e.*, inverse the role of \mathbf{X} and \mathbf{Y} with respect to the model of section 3.2.1. Then use the forward mapping function (3.11).

We respectively refer to these two strategies as *inverse mapping strategy* and *forward mapping strategy*. Let us now analyze their intrinsic difference.

Theorem 1 states that when none of the parameters are constrained, the joint distribution $p(\mathbf{X}, \mathbf{Y}; \boldsymbol{\theta})$ is a GMM. We immediately deduce the following corollary:

Corollary 1 *For the unconstrained GLLiM model, the inverse mapping strategy and the forward mapping strategy are strictly equivalent.*

Proof: Since both $\tilde{\boldsymbol{\theta}}$ and $\tilde{\boldsymbol{\theta}}^*$ maximizes the observed-data-log-likelihood of the same GMM, they can be mapped to the same set of GMM parameters using (3.7). The formulas mapping $\tilde{\boldsymbol{\theta}}$ to $\tilde{\boldsymbol{\theta}}^*$ are given by (3.10). Hence, the inverse mapping strategy and the forward mapping strategy are equivalent ■.

Importantly, Corollary 1 is only true if the GLLiM parameters are unconstrained. If, for instance, diagonal or isotropic constraints are added to covariance matrices $\{\boldsymbol{\Gamma}_k\}_{k=1}^K$ or $\{\boldsymbol{\Sigma}_k\}_{k=1}^K$, the forward and inverse mapping strategies are not equivalent. To see this, let's consider a practical case where $L = 2$, $D = 1000$. We assume that \mathbf{Y} depends on \mathbf{X} through a smooth function \mathbf{g} , *i.e.*, it lies on a L -dimensional manifold, and is corrupted by isotropic noise. We assume that \mathbf{g} is approximately piecewise-affine with $K = 10$ components. Under such hypothesis, if we use the inverse mapping strategy, it is natural to constrain the noise covariance matrices $\{\boldsymbol{\Sigma}_k\}_{k=1}^K$ in \mathbb{R}^D to be isotropic and equal for all k (see equation (3.1)). The dimension of $\tilde{\boldsymbol{\theta}}$ is then $\mathcal{D}(\tilde{\boldsymbol{\theta}}) = K(1+L+DL+L^2+D+1) = 30,080$. If we use the forward mapping strategy with the same constraints on $\{\boldsymbol{\Sigma}_k^*\}_{k=1}^K$, the dimension of $\tilde{\boldsymbol{\theta}}^*$ is $K(1+D+LD+D^2+L+1) = 10,030,040$. Accurately estimating such a large number of parameters is impossible in practice, as it would require a huge amount of training data. Note that if we use the forward mapping strategy with isotropic-equal constraints on $\{\boldsymbol{\Gamma}_k^*\}_{k=1}^K$ instead, we obtain the same parameter dimension $\mathcal{D}(\tilde{\boldsymbol{\theta}}^*) = 30,080$. However, this corresponds to fitting an isotropic Gaussian mixture model on the high-dimensional. In other words, it assumes that all the components of high-dimensional observations are independent *a priori*, whereas they all depend on the same low-dimensional variable \mathbf{X} through function \mathbf{g} , by hypothesis. Such a model would therefore have a very poor fit on manifold data.

In conclusion, the inverse mapping strategy combined with equal and isotropic (or diagonal) covariances $\{\boldsymbol{\Sigma}_k\}_{k=1}^K$ seems to be the most natural and most efficient way to deal with high-dimensional data presenting a locally-linear dependency on low-dimensional data and corrupted by isotropic (or diagonal) noise. This is typically the case when observations lie on a manifold, *e.g.*, the interaural features presented in Chapter 2. This scheme and model will be referred to as *probabilistic piecewise affine mapping* (PPAM).

The model underlying PPAM may in fact be viewed as an instance of the mixture of local experts (MLE) model introduced by [Xu 95], where each local transformation is affine. Although a closed form and efficient *expectation-maximization* (EM) algorithm exists for that instance (section 3.3.3), it has barely been used in the literature, to the best of our knowledge. In addition, as detailed in next section, PPAM has a nice geometrical interpretation as a piecewise-affine mapping technique for data lying on a manifold. This interpretation has not been studied in the context of MLE models. This is probably because MLE was designed as a probabilistic interpretation of neural networks, and has mostly been used with logistic, generalized linear, or binary functions [Peng 96, Avnimelech 99, Giacinto 00, Güler 05]. Moreover, a generalization of MLE referred to as hierarchical mixture of experts [Jordan 94, Huerta 03] is more often used in practice. We note that the use of these models in practical applications have gradually decreased over the past decade. More generally, the inverse mapping strategy seems to have been overlooked in the literature, although it presents several advantages for high-to-low dimensional regression with data lying on a manifold.

3.3.2 Geometrical interpretation

According to eq. (3.3), the probability of Z_n conditioned by \mathbf{x}_n writes:

$$p(Z_n = k | \mathbf{x}_n; \boldsymbol{\theta}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n; \mathbf{c}_k, \boldsymbol{\Gamma}_k)}{\sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n; \mathbf{c}_k, \boldsymbol{\Gamma}_k)}. \quad (3.13)$$

We can give a geometrical interpretation of this distribution by adding the following *volume equality* constraints to the model:

$$|\boldsymbol{\Gamma}_1| = \dots = |\boldsymbol{\Gamma}_K| \text{ and } \pi_1 = \dots = \pi_K = 1/K. \quad (3.14)$$

Under these constraints, the set of K regions of $\mathcal{X} \subseteq \mathbb{R}^L$ maximizing (3.13) for each k defines a Voronoi diagram of centroids $\{\mathbf{c}_k\}_{k=1}^K$, where the Mahalanobis distance $\|\cdot\|_{\boldsymbol{\Gamma}_k}$ is used instead of the Euclidean one. This corresponds to a compact probabilistic way of representing a general partitioning of the low-dimensional space into convex regions of equal volume. Fig. 3.1 compares partitionings obtained with or without the constraint, using the EM algorithm devised in next section on a toy datasets. The partitioning obtained with the volume equality constraint respects the symmetry of the data, contain balanced regions in terms of volume, and does not feature crossing between regions. On the other hand, the partitioning obtained without the constraint features several crossings, does not respects the symmetry of the data, and has unbalanced regions. In addition, experiments with these toy data tended to show that the algorithm converged much faster when adding the constraint than without. Although the partitionings obtained using the volume equality are visually better and easier to interpret in terms of piecewise affine mapping, no significant improvement was observed quantitatively in terms of mapping errors in later experiments.

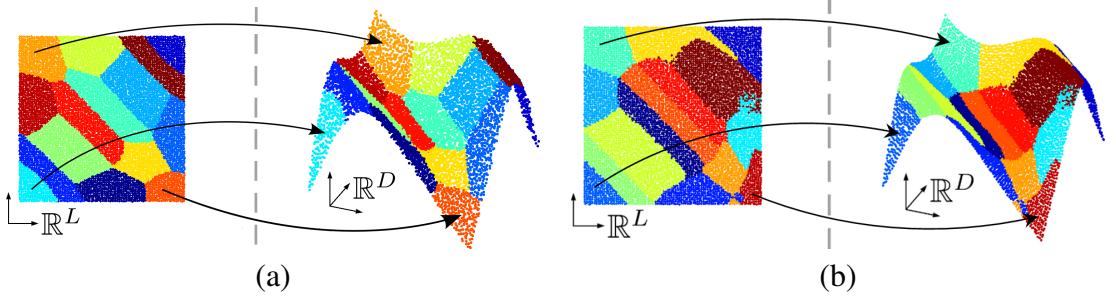


Figure 3.1: PPAM applied to a toy data set ($L = 2, D = 3, K = 15$) with (a) or without (b) the volume equality constraint (3.14). Colors encode regions maximizing the final posterior probabilities r_{kn}^∞ obtained after convergence of the algorithm, as defined in (3.17). Notice how PPAM automatically adjusts these regions (associated with affine transformations) to the geometry of the data.

3.3.3 Expectation-maximization inference for PPAM

The inference of PPAM can be done with a closed-form and efficient EM algorithm maximizing the observed-data log-likelihood $\log p(\{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^N; \boldsymbol{\theta})$ with respect to the model parameters:

$$\boldsymbol{\theta} = \left\{ \left\{ \boldsymbol{\Gamma}_k, \mathbf{c}_k, \mathbf{A}_k, \mathbf{b}_k \right\}_{k=1}^K, \boldsymbol{\Sigma} \right\}, \quad (3.15)$$

where

$$\boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, \dots, \sigma_D^2) \quad (3.16)$$

Posteriors at iteration i are defined by

$$r_{kn}^{(i)} = p(Z_n = k | \mathbf{x}_n, \mathbf{y}_n; \boldsymbol{\theta}^{(i-1)}) \quad (3.17)$$

and are computed in the E-step using (3.2), (3.3) and Bayes inversion. The M-step maximizes the expected complete-data log-likelihood $\mathbb{E}_{(Z|\mathbf{X}, \mathbf{Y}, \boldsymbol{\theta}^{(i)})}[\log p(\mathbf{X}, \mathbf{Y}, Z | \boldsymbol{\theta})]$. We obtain the following closed-form expressions for the parameters updates under the volume equality constraints (3.14):

$$\mathbf{c}_k^{(i)} = \sum_{n=1}^N \frac{r_{kn}^{(i)}}{\bar{r}_k^{(i)}} \mathbf{x}_n, \quad \boldsymbol{\Gamma}_k^{(i)} = \frac{\mathbf{S}_k^{(i)}}{|\mathbf{S}_k^{(i)}|^{\frac{1}{L}}} \sum_{j=1}^K \frac{\bar{r}_j^{(i)}}{N} |\mathbf{S}_j^{(i)}|^{\frac{1}{L}} \quad (3.18)$$

$$\mathbf{A}_k^{(i)} = \bar{\mathbf{Y}}_k^{(i)} \bar{\mathbf{X}}_k^{(i)\dagger}, \quad \mathbf{b}_k^{(i)} = \sum_{n=1}^N \frac{r_{kn}^{(i)}}{\bar{r}_k^{(i)}} (\mathbf{y}_n - \mathbf{A}_k^{(i)} \mathbf{x}_n), \quad (3.19)$$

$$\sigma_d^{2(i)} = \frac{1}{K} \sum_{k=1}^K \sum_{n=1}^N \frac{r_{kn}^{(i)}}{\bar{r}_k^{(i)}} (y_{dn} - \mathbf{a}_{dk}^{(i)\top} \mathbf{x}_n - b_{dk}^{(i)})^2, \quad (3.20)$$

where \dagger is the Moore-Penrose pseudo inverse operator, (\cdot, \cdot) denotes horizontal concatenation and:

$$\begin{aligned}\mathbf{S}_k^{(i)} &= \sum_{n=1}^N r_{kn}^{(i)} / \bar{r}_k^{(i)} (\mathbf{x}_n - \mathbf{c}_k^{(i)}) (\mathbf{x}_n - \mathbf{c}_k^{(i)})^\top \\ \bar{r}_k^{(i)} &= \sum_{k=1}^K r_{kn}^{(i)}, \quad \mathbf{A}_k^{(i)} = (\mathbf{a}_{1k}^{(i)}, \dots, \mathbf{a}_{Dk}^{(i)})^\top \\ \bar{\mathbf{X}}_k^{(i)} &= (r_{k1}^{(i)\frac{1}{2}} (\mathbf{x}_1 - \bar{\mathbf{x}}_k^{(i)}) \dots r_{kN}^{(i)\frac{1}{2}} (\mathbf{x}_N - \bar{\mathbf{x}}_k^{(i)})) \\ \bar{\mathbf{Y}}_k^{(i)} &= (r_{k1}^{(i)\frac{1}{2}} (\mathbf{y}_1 - \bar{\mathbf{y}}_k^{(i)}) \dots r_{kN}^{(i)\frac{1}{2}} (\mathbf{y}_N - \bar{\mathbf{y}}_k^{(i)})) \\ \bar{\mathbf{x}}_k^{(i)} &= \sum_{n=1}^N r_{kn}^{(i)} / \bar{r}_k^{(i)} \mathbf{x}_n, \quad \bar{\mathbf{y}}_k^{(i)} = \sum_{n=1}^N r_{kn}^{(i)} / \bar{r}_k^{(i)} \mathbf{y}_n.\end{aligned}$$

Initial posteriors $r_{kn}^{(0)}$ can be obtained either by estimating a K -GMM solely on \mathbf{X} or on joint data $[\mathbf{X}; \mathbf{Y}]$ where $[\cdot; \cdot]$ denotes vertical concatenation, and then go on with the M-step (3.18). Although the latter strategy may provide a better initialization than the former per Theorem 1, it is also much more computationally demanding when D is large (estimation of $K(D + L) \times (D + L)$ full rank covariance matrices).

3.4 Partially Latent Output Mapping: A Hybrid Model

3.4.1 Motivation

The PPAM model of previous section corresponds to the regression case, *i.e.*, a supervised GLLiM model where \mathbf{X} is *fully-observed*. In this section we present a new model referred to as *partially-latent-output mapping* (PLOM). It can be viewed as a generalization of supervised and unsupervised GLLiM, that allows for new hybrid models. While the input \mathbf{Y} remains fully observed, the output \mathbf{X} is the concatenation of an observed part \mathbf{T} and of an unobserved part \mathbf{W} , namely $\mathbf{X} = [\mathbf{T}; \mathbf{W}]$ where $[\cdot; \cdot]$ denotes the vertical concatenation of two column vectors. The graphical representations of supervised GLLiM models, unsupervised GLLiM models and PLOM are depicted in Figure 3.2.

PLOM has the potential of dealing with many applications, where the output can only be partially observed, either because it cannot be measured with appropriate sensors, or because it cannot be easily annotated. In other terms, it allows for some form of *slack*

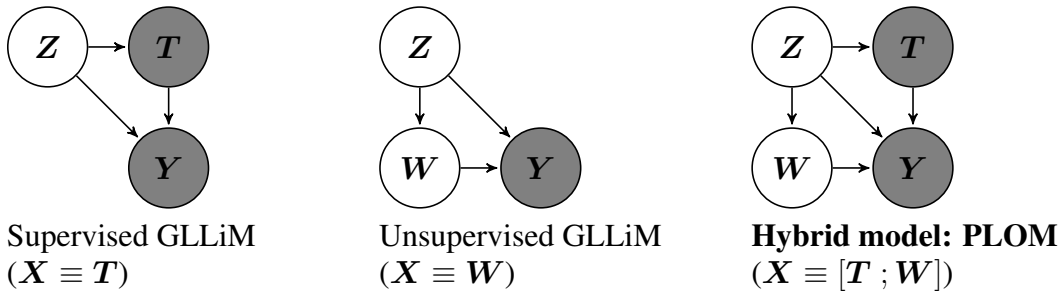


Figure 3.2: Graphical models. White means unobserved, gray means observed.

in the output variable by adding a few latent components to the otherwise observed ones. In order to motivate the need for such models, let us consider a few examples. Motion capture methods use regression to infer a map from high-dimensional visual data onto a small number of human joints involved in the particular motion that is being trained, *e.g.*, [Agarwal 04, Agarwal 06]. Nevertheless, the input data contain irrelevant information, such as lighting effects responsible for various artifacts, which aside from the fact that it is not relevant for the task at hand, is almost impossible to be properly modeled, quantified or even annotated. The recovered low-dimensional representation should also account for such phenomena that are unobservable.

In the field of planetology, hyper-spectral imaging is used to recover parameters associated with the physical properties of planet surfaces *e.g.*, [Bernard-Michel 09]. To this end, radiative transfer models have been developed, that link the chemical composition, the granularity, or the physical state, to the observed spectrum. They are generally used to simulate huge collections of spectra in order to perform the inversion of hyperspectral images [Douté 07]. As the required computing resources to generate such a database increases exponentially with the number of parameters, they are generally restricted to a small number of parameters, *e.g.* abundance and grain size of the main chemical components. Other parameters, such as those related to meteorological variability or the incidence angle of the spectrometer are not explicitly modeled and measured, in order to keep both the model and the database tractable.

Finally, in binaural sound source localization, the interaural feature vectors (see chapter 2) depend on the source position, whose locations and identities may be observed, and of reverberations, that are strongly dependent on the experimental conditions, and for which ground-truth data are barely available.

3.4.2 The PLOM model

The key idea is to treat \mathbf{X} as a *partially-latent* variable, namely $\mathbf{X} = \begin{bmatrix} \mathbf{T} \\ \mathbf{W} \end{bmatrix}$, where $\mathbf{T} \in \mathbb{R}^{L_t}$ is observed and $\mathbf{W} \in \mathbb{R}^{L_w}$ is latent ($L = L_t + L_w$). This simply means that the mapping's parameter estimation process uses observed pairs $\{\mathbf{y}_n, \mathbf{t}_n\}_{n=1}^N$ while it must also be constrained by the presence of the latent variable \mathbf{W} . This can be seen as a *latent-variable augmentation* of classical regression, where the observed realizations of \mathbf{Y} are affected by the unobserved variable \mathbf{W} . It can also be viewed as a variant of dimensionality reduction since the unobserved low-dimensional variable \mathbf{W} must be recovered from $\{\mathbf{y}_n, \mathbf{t}_n\}_{n=1}^N$. The decomposition of \mathbf{X} into observed and latent parts implies that some of the model parameters must be decomposed as well, namely \mathbf{c}_k , $\mathbf{\Gamma}_k$ and \mathbf{A}_k . Assuming the independence of \mathbf{T} and \mathbf{W} given \mathbf{Z} we write:

$$\mathbf{c}_k = \begin{bmatrix} \mathbf{c}_k^t \\ \mathbf{c}_k^w \end{bmatrix}, \mathbf{\Gamma}_k = \begin{bmatrix} \mathbf{\Gamma}_k^t & \mathbf{0} \\ \mathbf{0} & \mathbf{\Gamma}_k^w \end{bmatrix}, \mathbf{A}_k = \begin{bmatrix} \mathbf{A}_k^t & \mathbf{A}_k^w \end{bmatrix}. \quad (3.21)$$

It follows that (3.1) rewrites as

$$\mathbf{Y} = \sum_{k=1}^K \mathbb{I}(Z = k) (\mathbf{A}_k^t \mathbf{T} + \mathbf{A}_k^w \mathbf{W} + \mathbf{b}_k + \mathbf{E}_k) \quad (3.22)$$

or equivalently:

$$\mathbf{Y} = \sum_{k=1}^K \mathbb{I}(Z = k) (\mathbf{A}_k^t \mathbf{T} + \mathbf{b}_k + \mathbf{A}_k^w \mathbf{c}_k^w + \mathbf{E}'_k) \quad (3.23)$$

where \mathbf{E}'_k is distributed according to a zero-centered Gaussian with the following $D \times D$ covariance matrix

$$\Sigma'_k = \Sigma_k + \mathbf{A}_k^w \Gamma_k^w \mathbf{A}_k^{w\top}. \quad (3.24)$$

Considering realizations of variables \mathbf{T} and \mathbf{Y} , one may thus view PLOM as a supervised GLLiM model in which the noise covariance has an unconventional structure, namely (3.24), where $\mathbf{A}_k^w \Gamma_k^w \mathbf{A}_k^{w\top}$ is at most a rank- L_w matrix. When Σ_k is diagonal, this structure is that of factor analysis with at most L_w factors, and represents a flexible compromise between a full covariance with $O(D^2)$ parameters on one side, and a diagonal covariance with $O(D)$ parameters on the other side. Let us consider the isotropic case, *i.e.*, $\Sigma_k = \sigma_k^2 \mathbf{I}$, $\forall k$. We obtain the following three cases for the proposed model:

- $L_w = 0$: This is the fully supervised case. $\Sigma'_k = \Sigma_k$ and this corresponds to PPAM (section 3.3).
- $L_w = D$: Σ'_k takes the form of a general covariance matrix and we obtain the JGMM model (Theorem 1). JGMM is the most general GLLiM model and requires the estimation of K full covariance matrices of size $(D + L) \times (D + L)$. This model becomes over-parameterized and untractable when D is too large.
- $0 < L_w < D$: This corresponds to the PLOM model, and yields a large number of new regression models *in between* PPAM and JGMM.

As it will be experimentally shown in section 3.6.2, in practical cases where the output variable is only partially observed during training, PLOM yields better results than PPAM, GMM and other existing fully supervised regression techniques.

3.4.3 Connection to existing methods

A number of existing methods can be seen as particular instances of PLOM with specific constraints, where either L_t or L_w is null. This is summarized in table 3.1, and detailed in this section.

At least 3 regression models can be viewed as instances of PLOM where $L_w = 0$. As proven in section 3.2.2, JGMM [Kain 98] corresponds to the case of unconstrained parameters. When using equal and diagonal or isotropic covariance matrices $\{\Sigma_k\}_{k=1}^K$ we obtain the PPAM model, presented in section 3.3. Mixtures of linear regressors (MLR)

Mapping Methods		\mathbf{c}_k	$\mathbf{\Gamma}_k$	π_k	\mathbf{A}_k	\mathbf{b}_k	$\mathbf{\Sigma}_k$	L_t	L_w	K
MLR	[de Veaux 89]	$\mathbf{0}_L$	$\infty \mathbf{I}_L$	-	-	-	iso.+eq.	-	0	-
JGMM	[Kain 98]	-	-	-	-	-	-	-	0	-
PPAM	[Deleforge 12a]	-	eq.	eq.	-	-	diag.+eq.	-	0	-
GTM	[Bishop 98]	fixed	$\mathbf{0}_L$	eq.	eq.	$\mathbf{0}_D$	iso.+eq.	0	-	-
PPCA	[Tipping 99b]	$\mathbf{0}_L$	\mathbf{I}_L	-	-	-	iso.	0	-	1
MPPCA	[Tipping 99a]	$\mathbf{0}_L$	\mathbf{I}_L	-	-	-	iso.	0	-	-
MFA	[Ghahramani 96]	$\mathbf{0}_L$	\mathbf{I}_L	-	-	-	diag.	0	-	-
PCCA	[Bach 05]	$\mathbf{0}_L$	\mathbf{I}_L	-	-	-	block	0	-	1
RCA	[Kalaitzis 12]	$\mathbf{0}_L$	\mathbf{I}_L	-	-	-	fixed	0	-	1

Table 3.1: PLOM parameter constraints recovering different existing methods. The first 3 rows are supervised GLLiM methods ($L_w = 0$, Fig. 3.2(a)), the last 6 are unsupervised GLLiM methods ($L_t = 0$, Fig. 3.2(b)). A value means that parameters are fixed to this value and not estimated, *iso.* means isotropic, *diag.* means diagonal, *eq.* means equal for all k , *|eq.|* means equal determinants for all k , *fixed* means fixed to an arbitrary value and not estimated, *block* means block-diagonal and - means not constrained.

[de Veaux 89] can be viewed as a degenerate cases of PLOM where the covariances $\{\mathbf{\Gamma}_k\}_{k=1}^K$ are set to $\Omega \mathbf{I}_L$, where $\Omega \rightarrow \infty$, i.e., there is no prior on \mathbf{X} . One can verify that with such constraint, the forward conditional expectation (3.11) boils down to a sum of K affine transformations weighted by π_k . The inverse conditional expectation (3.12) boils down to a sum of K affine projections to the lower-dimensional space weighted by π_k . Affine projections are obtained from the pseudo-inverse of affine transformation matrices $\{\mathbf{A}_k\}_{k=1}^K$.

Several dimensionality reduction techniques can be viewed as instances of PLOM where $L_t = 0$. This is the case for probabilistic principal component analysis (PPCA) [Tipping 99b] and its mixture version (MPPCA) [Tipping 99a] where the covariances $\{\mathbf{\Sigma}_k\}_{k=1}^K$ are isotropic. Mixture of factor analyzers (MFA) [Ghahramani 96] corresponds to diagonal covariances, probabilistic canonical correlation analysis (PCCA) [Bach 05] to block-diagonal covariances, and residual component analysis (RCA) [Kalaitzis 12] to fixed (not estimated) covariances. A more hidden link also exists between unsupervised GLLiM and the dimensionality reduction technique *generative topographic mapping* (GTM) introduced by [Bishop 98]. In GTM, observations in the feature space \mathbb{R}^D are expressed as the linear transformation of a function $\phi : \mathbb{R}^{L'} \rightarrow \mathbb{R}^L$ of hidden latent variables in $\mathbb{R}^{L'}$ plus some noise. This writes $\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{E}$ where $\mathbf{X} = \phi(\mathbf{V}) \in \mathbb{R}^L$, $\mathbf{V} \in \mathbb{R}^{L'}$, $\mathbf{A} \in \mathbb{R}^{D \times L}$, and \mathbf{E} is a D -variate Gaussian noise variable with 0 mean and isotropic covariance matrix $\sigma^2 \mathbf{I}_D$. The components $\phi_1 \dots \phi_L$ of ϕ are seen as a set of L basis functions, chosen as Gaussians regularly spaced on a grid in $\mathbb{R}^{L'}$. The latent variable \mathbf{V} is assumed to follow a mixture of K Diracs, centered on the nodes of a regular grid $\{\mathbf{m}_1 \dots \mathbf{m}_K\}$ in $\mathbb{R}^{L'}$. Consequently, the variable $\mathbf{X} = \phi(\mathbf{V})$ also follows a mixture of Diracs in \mathbb{R}^L , centered at $\{\phi(\mathbf{m}_1) \dots \phi(\mathbf{m}_K)\}$. In the GLLiM framework, this corresponds to fixing \mathbf{c}_k to $\phi(\mathbf{m}_k)$, and $\mathbf{\Gamma}_k$ to $\epsilon^2 \mathbf{I}_L$ where $\epsilon \rightarrow 0$ for all k . In other words, diracs are modeled by degenerate isotropic Gaussians with null covariances. In addition, $\{\mathbf{b}_k\}_{k=1}^K$ are all set to 0, $\{\mathbf{A}_k\}_{k=1}^K$ are constrained to be all equal to \mathbf{A} and π_k

to $1/K$. The only free parameters to estimate are hence \mathbf{A} and σ^2 . For a given \mathbf{y} , with all the mentioned constraints and $\epsilon \rightarrow 0$, the inverse conditional density given in (3.12) becomes a mixture of Diracs in \mathbb{R}^L centered at $\{\phi(\mathbf{m}_1) \dots \phi(\mathbf{m}_K)\}$ and weighted by posterior probabilities $r_k(\mathbf{y})$ where

$$r_k(\mathbf{y}) = p(Z = k | \mathbf{y}; \boldsymbol{\theta}) = \frac{\mathcal{N}(\mathbf{y}; \mathbf{A}\phi(\mathbf{m}_k); \sigma^2 \mathbf{I}_D)}{\sum_{j=1}^K \mathcal{N}(\mathbf{y}; \mathbf{A}\phi(\mathbf{m}_j); \sigma^2 \mathbf{I}_D)}. \quad (3.25)$$

We deduce that the latent variable \mathbf{V} associated to a given \mathbf{y} also follows a Dirac mixture in $\mathbb{R}^{L'}$ centered at \mathbf{m}_k and weighted by $r_k(\mathbf{y})$. This is exactly the result of GTM [Bishop 98].

While unifying these methods, PLOM enables a wide range of generalizations corresponding to $L_t > 0$ and $L_w > 0$, *i.e.*, a partially latent output. To the best of our knowledge, there is no method achieving high-to-low dimensional regression with partially latent output in the current literature. It is worth noting that an appropriate choice of kernel function for Gaussian process latent variable models (GPLVM) [Lawrence 05] allows to account for a partially observed low-dimensional variable \mathbf{X} . This was notably studied in [Fusi 12]. However, as explained in [Lawrence 05], GPLVM only leads to a mapping from \mathbf{X} to \mathbf{Y} . This mapping is non-invertible due to the non-linearity of the kernel functions used in practice, as explained in section 3.1.3. Hence, GPLVM may allow for partially latent *input* regression, but not for partially latent *output* regression.

3.5 Expectation-maximization inference for PLOM

3.5.1 Two data augmentation schemes

In the general PLOM model, there are two sets of hidden variables, $Z_{1:N} = \{Z_n\}_{n=1}^N$ and $\mathbf{W}_{1:N} = \{\mathbf{W}_n\}_{n=1}^N$, associated with the observed training data $(\mathbf{y}, \mathbf{t})_{1:N} = \{\mathbf{y}_n, \mathbf{t}_n\}_{n=1}^N$. Two augmentation schemes arise naturally. The first scheme (referred to as general PLOM-EM) consists of augmenting the observed data with both variables $(Z, \mathbf{W})_{1:N}$ while the second scheme (referred to as marginal PLOM-EM) consists of integrating out the continuous variables $\mathbf{W}_{1:N}$ previous to data augmentation with the discrete variables $Z_{1:N}$. The difference between these two schemes is in the amount of missing information and this may be of interest considering the well-known fact that the convergence rates of EM procedures are determined by the portion of missing information in complete data. To accelerate standard EM algorithms it is natural to decrease the amount of missing data, but the practical computational gain is effective only on the premise that the corresponding M-step can be solved efficiently.

The general PLOM-EM algorithm, described in section 3.5.3 leads to closed-form expressions for a wide range of constraints onto the covariance matrices $\{\boldsymbol{\Gamma}_k\}_{k=1}^K$ and $\{\boldsymbol{\Sigma}_k\}_{k=1}^K$. Moreover, the algorithm can be applied to both supervised ($L_w = 0$) and unsupervised ($L_t = 0$) GLLiM models. Hence, it can be viewed as a generalization of a number of EM inference techniques for regression, *e.g.*, MLR, MLE, JGMM, GTM, or

for dimensionality reduction, *e.g.*, PPCA, MPPCA, MFA and RCA (see section 3.4.3. The marginal PLOM-EM algorithm, described in section 3.5.4, is less general. Nevertheless, it is of interest because it provides both an algorithmic insight into the PLOM model as well as a natural initialization strategy for the general algorithm.

3.5.2 A note on non-identifiability

Notice that the means $\{\mathbf{c}_k^w\}_{k=1}^K$ and covariance matrices $\{\mathbf{\Gamma}_k^w\}_{k=1}^K$ must be fixed to avoid non-identifiability. Indeed, changing their values respectively corresponds to shifting and scaling the unobserved variables $\mathbf{W}_{1:N}$ in \mathbb{R}^{L_w} , which can be compensated by changing local affine transformation parameters $\{\mathbf{A}_k^w\}_{k=1}^K$ and $\{\mathbf{b}_k\}_{k=1}^K$. The same issue is observed in all latent variable models used for dimensionality reduction and is always solved by fixing these parameters. In GTM [Bishop 98] the means are spread on a regular grid and the covariance matrices are set to $\mathbf{0}$ (Dirac functions), while in MPPCA [Tipping 99a] and MFA [Ghahramani 96] all means and covariance matrices are respectively set to $\mathbf{0}_{L_w}$ and \mathbf{I}_{L_w} . The latter option will be used in all experiments (Section 3.6.2, 3.6.3, 3.6.4 and 3.6.5), but for the sake of generality, the general PLOM-EM algorithm is derived for any fixed means and covariance matrices in section 3.5.3.

3.5.3 The general PLOM-EM algorithm

We present here an EM algorithm for the most general case of PLOM where both L_w and L_t are strictly positive. We call this version *general*, because it may be used with a large choice of constraints on covariance matrices $\{\mathbf{\Sigma}_k\}_{k=1}^K$, which is not the case for the marginal PLOM-EM presented in the next section. Considering the complete data, with $(\mathbf{Y}, \mathbf{T})_{1:N}$ being the observed variables and $(Z, \mathbf{W})_{1:N}$ being the missing ones, the corresponding EM algorithm consists of estimating the parameter vector $\boldsymbol{\theta}^{(i+1)}$ that maximizes the following objective function, given the current parameter vector $\boldsymbol{\theta}^{(i)}$:

$$\boldsymbol{\theta}^{(i+1)} = \arg \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)}), \quad (3.26)$$

with:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)}) = \mathbb{E}_{r_{W,Z}^{(i+1)}} [\log p((\mathbf{y}, \mathbf{t}, \mathbf{W}, Z)_{1:N}; \boldsymbol{\theta}) | (\mathbf{y}, \mathbf{t})_{1:N}; \boldsymbol{\theta}^{(i)}]. \quad (3.27)$$

where $\mathbb{E}_{r_{W,Z}^{(i+1)}}$ denotes the expectation with respect to the current posterior distribution $r_{W,Z}^{(i+1)} = p((\mathbf{W}, Z)_{1:N} | (\mathbf{y}, \mathbf{t})_{1:N}; \boldsymbol{\theta}^{(i)})$. Using that $\mathbf{W}_{1:N}$ and $\mathbf{T}_{1:N}$ are independent conditionally on $Z_{1:N}$ and that $\{\mathbf{c}_k^w\}_{k=1}^K$ and $\{\mathbf{\Gamma}_k^w\}_{k=1}^K$ are fixed, maximizing Q is then equivalent to maximizing the following expression:

$$\mathbb{E}_{r_Z^{(i+1)}} [\mathbb{E}_{r_{W|Z}^{(i+1)}} [\log p(\mathbf{y}_{1:N} | (\mathbf{t}, \mathbf{W}, Z)_{1:N}; \boldsymbol{\theta})] + \log p((\mathbf{t}, Z)_{1:N}; \boldsymbol{\theta})] \quad (3.28)$$

where $r_Z^{(i+1)}$ and $r_{W|Z}^{(i+1)}$ respectively denote the posterior distributions

$$p(Z_{1:N} | (\mathbf{y}, \mathbf{t})_{1:N}; \boldsymbol{\theta}^{(i)}) \quad \text{and} \quad p(\mathbf{W}_{1:N} | (\mathbf{y}, \mathbf{t}, Z)_{1:N}; \boldsymbol{\theta}^{(i)}).$$

It follows that the E-step splits into the following **E-W** and **E-Z** steps. For the sake of readability, the current iteration superscript $(i + 1)$ is replaced with a tilde, *e.g.*, $\mu^{(i+1)}$ is replaced with $\tilde{\mu}$.

E-W-step: The posterior probability $\tilde{r}_{W|Z}$ given previous parameters estimates is fully defined by determining for all n and all k distribution $p(\mathbf{w}_n | Z_n = k, \mathbf{t}_n, \mathbf{y}_n; \boldsymbol{\theta}^{(i)})$ which can be shown to be Gaussian with mean and covariance matrix denoted by $\tilde{\boldsymbol{\mu}}_{nk}^w$ and $\tilde{\mathbf{S}}_k^w$ where:

$$\tilde{\boldsymbol{\mu}}_{nk}^w = \tilde{\mathbf{S}}_k^w (\mathbf{A}_k^{w(i)\top} \boldsymbol{\Sigma}_k^{(i)-1} (\mathbf{y}_n - \mathbf{A}_k^{t(i)} \mathbf{t}_n - \mathbf{b}_k^{(i)}) + \boldsymbol{\Gamma}_k^{w-1} \mathbf{c}_k^w) \quad (3.29)$$

$$\tilde{\mathbf{S}}_k^w = (\boldsymbol{\Gamma}_k^{w-1} + \mathbf{A}_k^{w(i)\top} \boldsymbol{\Sigma}_k^{(i)-1} \mathbf{A}_k^{w(i)})^{-1} \quad (3.30)$$

Conditionally to $Z_n = k$, equation (3.23) shows that this step amounts to a factor analysis step. Indeed, we recover standard formula for the posterior over latent factors where the observations are replaced by the *current residuals* $(\mathbf{y}_n - \mathbf{A}_k^{t(i)} \mathbf{t}_n - \mathbf{b}_k^{(i)})$.

E-Z-step: The posterior probability \tilde{r}_Z is defined by:

$$\tilde{r}_{nk} = p(Z_n = k | \mathbf{t}_n, \mathbf{y}_n; \boldsymbol{\theta}^{(i)}) = \frac{\pi_k^{(i)} p(\mathbf{y}_n, \mathbf{t}_n | Z_n = k; \boldsymbol{\theta}^{(i)})}{\sum_{j=1}^K \pi_j^{(i)} p(\mathbf{y}_n, \mathbf{t}_n | Z_n = j; \boldsymbol{\theta}^{(i)})}$$

for all n and all k where

$$p(\mathbf{y}_n, \mathbf{t}_n | Z_n = k; \boldsymbol{\theta}^{(i)}) = p(\mathbf{y}_n | \mathbf{t}_n, Z_n = k; \boldsymbol{\theta}^{(i)}) p(\mathbf{t}_n | Z_n = k; \boldsymbol{\theta}^{(i)}).$$

The second term is equal to $\mathcal{N}(\mathbf{t}_n; \mathbf{c}_k^w; \boldsymbol{\Gamma}_k^w)$ by (3.3) and (3.21) while it is clear from (3.23) that

$$p(\mathbf{y}_n | \mathbf{t}_n, Z_n = k; \boldsymbol{\theta}^{(i)}) = \mathcal{N}(\mathbf{y}_n; \mathbf{A}_k^{(i)} [\mathbf{t}_n; \mathbf{c}_k^w] + \mathbf{b}_k^{(i)}, \mathbf{A}_k^{w(i)} \boldsymbol{\Gamma}_k^w \mathbf{A}_k^{w(i)\top} + \boldsymbol{\Sigma}_k^{(i)}).$$

The maximization of Q can then be performed using the posterior probabilities \tilde{r}_{nk} and sufficient statistics $\tilde{\boldsymbol{\mu}}_{nk}^w$ and $\tilde{\mathbf{S}}_k^w$. We use the following notation, $\bar{r}_k = \sum_{n=1}^N \tilde{r}_{nk}$ and $\tilde{\mathbf{x}}_{nk} = [\mathbf{t}_n; \tilde{\boldsymbol{\mu}}_{nk}^w] \in \mathbb{R}^L$. It can be easily seen in the decomposition (3.28) of Q , that the M-step can be divided into two separated steps. First, the updates of parameters $\tilde{\pi}_k$, $\tilde{\mathbf{c}}_k^t$ and $\tilde{\boldsymbol{\Gamma}}_k^t$ correspond to those of a standard Gaussian mixture model on $\mathbf{T}_{1:N}$ so that we get straightforwardly:

$$\mathbf{M}\text{-GMM-step: } \tilde{\pi}_k = \frac{\bar{r}_k}{N}, \tilde{\mathbf{c}}_k^t = \sum_{n=1}^N \frac{\tilde{r}_{kn}}{\bar{r}_k} \mathbf{t}_n, \tilde{\boldsymbol{\Gamma}}_k^t = \sum_{n=1}^N \frac{\tilde{r}_{kn}}{\bar{r}_k} (\mathbf{t}_n - \tilde{\mathbf{c}}_k^t)(\mathbf{t}_n - \tilde{\mathbf{c}}_k^t)^\top.$$

Second, the updating of the mapping parameters $\{\mathbf{A}_k, \mathbf{b}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ is also in closed-form:

$$\mathbf{M}\text{-mapping-step: } \tilde{\mathbf{A}}_k = \tilde{\mathbf{Y}}_k \tilde{\mathbf{X}}_k^\top (\tilde{\mathbf{S}}_k^x + \tilde{\mathbf{X}}_k \tilde{\mathbf{X}}_k^\top)^{-1} \text{ where } \tilde{\mathbf{S}}_k^x = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{S}}_k^w \end{bmatrix}, \tilde{\mathbf{X}}_k = \frac{1}{\bar{r}_k} \left[\tilde{r}_{1k}^{\frac{1}{2}} (\tilde{\mathbf{x}}_{1k} - \tilde{\mathbf{x}}_k) \dots \tilde{r}_{Nk}^{\frac{1}{2}} (\tilde{\mathbf{x}}_{Nk} - \tilde{\mathbf{x}}_k) \right], \tilde{\mathbf{Y}}_k = \frac{1}{\bar{r}_k} \left[\tilde{r}_{1k}^{\frac{1}{2}} (\mathbf{y}_1 - \tilde{\mathbf{y}}_k) \dots \tilde{r}_{Nk}^{\frac{1}{2}} (\mathbf{y}_N - \tilde{\mathbf{y}}_k) \right],$$

$$\tilde{\mathbf{x}}_k = \sum_{n=1}^N \frac{\tilde{r}_{kn}}{\bar{r}_k} \tilde{\mathbf{x}}_{nk} \text{ and } \tilde{\mathbf{y}}_k = \sum_{n=1}^N \frac{\tilde{r}_{kn}}{\bar{r}_k} \mathbf{y}_n. \text{ When } L_w = 0 \text{ then } \tilde{\mathbf{S}}_k^x = \mathbf{0} \text{ and the above}$$

expression of $\tilde{\mathbf{A}}_k$ is that of standard linear regression from $\{\mathbf{t}_n\}_{n=1}^N$ to $\{\mathbf{y}_n\}_{n=1}^N$ weighted by $\{\tilde{r}_{nk}\}_{n=1}^N$, *i.e.*, equation (3.19) in PPAEM's M-step. When $L_t = 0$ then $\tilde{\mathbf{S}}_k = \tilde{\mathbf{S}}_k^w$ and we obtain the principal components update of the PPCA EM algorithm [Tipping 99b]. The intercept parameter updates as:

$$\tilde{\mathbf{b}}_k = \sum_{n=1}^N \frac{\tilde{r}_{kn}}{\tilde{r}_k} (\mathbf{y}_n - \tilde{\mathbf{A}}_k \tilde{\mathbf{x}}_{nk}) \quad (3.31)$$

and we obtain the following expression for $\tilde{\Sigma}_k$:

$$\tilde{\Sigma}_k = \tilde{\mathbf{A}}_k \tilde{\mathbf{S}}_k^w \tilde{\mathbf{A}}_k^{\top} + \sum_{n=1}^N \frac{\tilde{r}_{kn}}{\tilde{r}_k} (\mathbf{y}_n - \tilde{\mathbf{A}}_k \tilde{\mathbf{x}}_{nk} - \tilde{\mathbf{b}}_k) (\mathbf{y}_n - \tilde{\mathbf{A}}_k \tilde{\mathbf{x}}_{nk} - \tilde{\mathbf{b}}_k)^{\top} \quad (3.32)$$

Note that the M-mapping-step formulas can be seen as standard weighted affine regression formula after *imputation* of the missing variables \mathbf{w}_n by their mean values $\tilde{\boldsymbol{\mu}}_{nk}^w$ via the definition of $\tilde{\mathbf{x}}_{nk}$. As such a direct imputation by the mean necessarily underestimates the variance, the above formula also contains an additional term involving the variance $\tilde{\mathbf{S}}_k^w$ of the missing data.

Formulas are given for unconstrained parameters, but can be straightforwardly adapted to different constraints. For instance, if $\{\mathbf{M}_k\}_{k=1}^K \subset \mathbb{R}^{P \times P}$ are solutions for unconstrained covariance matrices $\{\Sigma_k\}_{k=1}^K$ or $\{\Gamma_k\}_{k=1}^K$, then solutions with diagonal (diag), isotropic (iso) and/or equal for all k (eq) constraints are respectively given by $\mathbf{M}_k^{\text{diag}} = \text{diag}(\mathbf{M}_k)$, $\mathbf{M}_k^{\text{iso}} = \frac{1}{P} \text{tr}(\mathbf{M}_k) \mathbf{I}_P$ and $\mathbf{M}^{\text{eq}} = \sum_{k=1}^K \tilde{\pi}_k \mathbf{M}_k$.

Initialization: In general, EM algorithms are known to be quite sensitive to initialization and may converge to undesired local maxima of the likelihood when initialized inappropriately. Initialization can either be done by choosing a set of parameter values and go on with the E-step, or choosing a set of posterior probabilities and go on with the M-step. The general PLOM-EM algorithm however, is such that there is no straightforward way of choosing a complete set of initial posteriors $(r_{nk}^{(0)}, \boldsymbol{\mu}_{nk}^{w(0)})$ and $\mathbf{S}_k^{w(0)}$ for all n, k) or a complete set of initial parameters $\boldsymbol{\theta}^{(0)}$ including all the local affine transformations. This issue is addressed by deriving a *marginal* variant of the above described algorithm, in which latent variables $\mathbf{W}_{1:N}$ are integrated out, leaving only the estimation of posteriors r_Z in the E-step. Full details on this variant are given in section 3.5.4. As explained there, this variant is much easier to initialize but has closed-form steps only if covariance matrices $\{\Sigma_k\}_{k=1}^K$ are isotropic and distinct. In practice, we thus run one iteration of the marginal variant to obtain a set of initial parameters $\boldsymbol{\theta}^{(0)}$ and go on until convergence with the general PLOM-EM described.

3.5.4 The marginal PLOM-EM algorithm

By marginalizing out the hidden variables $\mathbf{W}_{1:N}$, we obtain a different EM algorithm than the one presented in section 3.5.3 with only hidden variables $\mathbf{Z}_{1:N}$. For a clearer connection with standard procedures, we assume here as specified earlier that $\mathbf{c}_k^w = \mathbf{0}_{L_w}$ and

$\Gamma_k^w = \mathbf{I}_{L_w}$. The **E-W-step** disappears while the **E-Z-step** and the following updating of π_k , \mathbf{c}_k^t and Γ_k^t in the **M-GMM-step** are exactly the same as in section 3.5.3. However, the marginalization of $\mathbf{W}_{1:N}$ leads to a clearer separation between the regression parameters \mathbf{A}_k^t and \mathbf{b}_k (**M-regression-step**) and the other parameters \mathbf{A}_k^w and Σ_k (**M-residual-step**). This can be seen straightforwardly from equation (3.23) which shows that after marginalizing \mathbf{W} , the model parameters separate into a standard regression part $\mathbf{A}_k^t \mathbf{t}_n + \mathbf{b}_k$ for which standard estimators do not involve the noise variance and a PPCA-like part, on the regression residuals $\mathbf{y}_n - \tilde{\mathbf{A}}_k^t \mathbf{t}_n - \tilde{\mathbf{b}}_k$, in which the non standard noise covariance $\Sigma_k + \mathbf{A}_k^w \mathbf{A}_k^{w\top}$ is typically dealt with by adding a latent variable \mathbf{W} .

The algorithm is therefore made of the **E-Z-step** and **M-GMM-step** detailed in 3.5.3 and the following additional M-steps:

M-regression-step: The \mathbf{A}_k^t and \mathbf{b}_k parameters are obtained using standard weighted affine regression from $\{\mathbf{t}_n\}_{n=1}^N$ to $\{\mathbf{y}_n\}_{n=1}^N$ with weights \tilde{r}_{nk} , *i.e.*,

$$\tilde{\mathbf{A}}_k^t = \tilde{\mathbf{Y}}_k \tilde{\mathbf{T}}_k^\top (\tilde{\mathbf{T}}_k \tilde{\mathbf{T}}_k^\top)^{-1}, \quad \tilde{\mathbf{b}}_k = \sum_{n=1}^N \frac{\tilde{r}_{kn}}{\tilde{r}_k} (\mathbf{y}_n - \tilde{\mathbf{A}}_k^t \mathbf{t}_n) \quad (3.33)$$

with $\tilde{\mathbf{T}}_k = \frac{1}{\sqrt{\tilde{r}_k}} \left[\sqrt{\tilde{r}_{1k}} (\mathbf{t}_1 - \tilde{\mathbf{t}}_k) \dots \sqrt{\tilde{r}_{Nk}} (\mathbf{t}_N - \tilde{\mathbf{t}}_k) \right]$ and $\tilde{\mathbf{t}}_k = \sum_{n=1}^N \frac{\tilde{r}_{kn}}{\tilde{r}_k} \mathbf{t}_n$.

M-residual-step: Optimal values for \mathbf{A}_k^w and Σ_k are obtained by minimization of the following criterion:

$$Q_k(\Sigma_k, \mathbf{A}_k^w) = -\frac{1}{2} \left(\log |\Sigma_k + \mathbf{A}_k^w \mathbf{A}_k^{w\top}| + \sum_{n=1}^N \mathbf{u}_{kn}^\top (\Sigma_k + \mathbf{A}_k^w \mathbf{A}_k^{w\top})^{-1} \mathbf{u}_{kn} \right) \quad (3.34)$$

where $\mathbf{u}_{kn} = \sqrt{\tilde{r}_{nk}/\tilde{r}_k} (\mathbf{y}_n - \tilde{\mathbf{A}}_k^t \mathbf{t}_n - \tilde{\mathbf{b}}_k)$. Vectors $\{\mathbf{u}_{kn}\}_{n=1}^N$ can be seen as the *residuals* of the k -th local affine transformation. No closed-form solution exists in the general case. A first option is to make use of an inner loop such as a gradient descent technique, or to consider Q_k as the new target observed-data likelihood and use an inner EM corresponding to the general EM described in previous section with $L_t = 0$ and $K = 1$.

However, in the particular case $\Sigma_k = \sigma_k^2 \mathbf{I}_D$, we can afford a standard EM as it connects to probabilistic PCA (PPCA) [Tipping 99b]. Indeed, one may notice that Q_k has then exactly the same form as the observed-data log-likelihood in PPCA, with parameters $(\sigma_k^2, \mathbf{A}_k^w)$ and observations $\{\mathbf{u}_{kn}\}_{n=1}^N$. Denoting by $\mathbf{C}_k = \frac{1}{N} \sum_{n=1}^N \mathbf{u}_{kn} \mathbf{u}_{kn}^\top$ the $D \times D$ sample *residual covariance matrix* and $\lambda_{1k} > \dots > \lambda_{Dk}$ its eigenvalues in decreasing order, we can therefore use the key result of [Tipping 99b] to see that a global maximum of Q_k is obtained for

$$\tilde{\mathbf{A}}_k^w = \mathbf{U}_k (\mathbf{\Lambda}_k - \sigma_k^2 \mathbf{I}_{L_w})^{1/2}, \quad \text{and} \quad \tilde{\sigma}_k^2 = \frac{\sum_{d=L_w+1}^D \lambda_{dk}}{D - L_w} \quad (3.35)$$

where \mathbf{U}_k denotes the $D \times L_w$ matrix whose column vectors are the first eigenvectors of \mathbf{C}_k and $\mathbf{\Lambda}_k$ is a $L_w \times L_w$ diagonal matrix containing the corresponding first eigenvalues. The hybridity of PLOM between regression and dimensionality reduction models is

striking in this variant, as it alternates between a mixture of Gaussians step, a local linear regression step and a local linear dimensionality reduction step on residuals. This variant is also much easier to initialize as a set of initial posterior values $\{r_{nk}^{(0)}\}_{n=1,k=1}^{N,K}$ can be obtained either by estimating a K -GMM solely on \mathbf{T} or on joint data $[\mathbf{T}; \mathbf{Y}]$, and then go on with the M-step (see end of section 3.3.3). On the other hand, due to the costly eigenvalue decomposition at each step it turned out to be slower than the PLOM-EM algorithm described in section 3.5.3, while being less general. We thus use the marginal PLOM-EM algorithm as an efficient initialization procedure for the general one.

3.6 Experiments and Results

3.6.1 Evaluation methodology

In this section, we evaluate the performance of the PPAM algorithm presented in section 3.3 and the PLOM algorithm presented in section 3.5 on high- to low-dimensional regression tasks. To do so, the algorithms are tested on 4 different datasets. In section 3.6.2 we inverse high-dimensional functions using synthetic data. In section 3.6.3 we retrieve pose or light information from face images. In section 3.6.4 we recover some physical properties of the Mars surface from hyperspectral images. In section 3.6.5 we localize a white-noise sound source using mean interaural feature vectors obtained from binaural recordings (see chapter 2).

For each of these 4 datasets, we consider two situations: i) the output data is completely observed and ii) The output data is only partially observed during training. While PPAM and PLOM will show to yield similar performance in the first situation, the second situation allows to highlight the prominent advantage of PLOM over standard regression methods in applications where the output can only be partially annotated. In each case, several PLOM models are tested, corresponding to different values of L_w . They are denoted PLOM- L_w . Recall that PPAM is actually a particular instance of PLOM where the dimensionality of the latent-output L_w is set to 0, *i.e.*, PPAM \equiv PLOM-0.

In all tasks considered, N observed training couples $\{(\mathbf{t}_n, \mathbf{y}_n)\}_{n=1}^N$ are used to obtain a set of parameters with the PPAM and PLOM algorithms. Then, we use the inverse mapping formula (3.12) to compute an estimate $\hat{\mathbf{t}}'$ given a test observation \mathbf{y}' . This is repeated for N' tests observations $\{\mathbf{y}'_n\}_{n=1}^{N'}$. The training and the test sets are disjoint in all experiments.

PPAM and PLOM are also compared to three existing regression techniques, namely joint GMM (JGMM) [Qiao 09] which is equivalent to PLOM with $L_w \geq D$ (see section 3.4.2), *sliced inverse regression* (SIR) [Li 91] and *multivariate relevance vector machine* (RVM) [Thayananthan 06]. SIR is used with one (SIR-1) or two (SIR-2) principal axes for dimensionality reduction, 20 slices (the number of slices is known to have very little influence on the results), and polynomial regression of order three (higher orders did not show significant improvements in experiments). SIR quantizes the low-dimensional data

Table 3.2: Average (Avg), standard deviation (Std) and percentage of extreme values (Ex) of the absolute error obtained with different methods for function α inversion and interpolation.

Method	α		
	Avg	Std	Ex
JGMM	2.07	2.38	24.7
SIR-1	1.43	1.21	8.78
SIR-2	0.73	0.87	2.54
RVM	0.65	0.53	0.10
PPAM	0.19	0.20	0.00
PLOM-1	0.22	0.23	0.00
PLOM-2	0.22	0.22	0.00

\mathbf{X} into *slices* or clusters which in turn induces a quantization of the \mathbf{Y} -space. Each \mathbf{Y} -slice (all points \mathbf{y}_n that map to the same \mathbf{X} -slice) is then replaced with its mean and PCA is carried out on these means. The resulting dimensionality reduction is then informed by \mathbf{X} values through the preliminary slicing. RVM [Thayananthan 06] may be viewed as a multivariate probabilistic formulation of *support vector regression* [Smola 04]. As all kernel methods, it critically depends on the choice of a kernel function. Using the authors' freely available code¹, we ran preliminary tests to determine an optimal kernel choice for each dataset considered. We tested 14 kernel types with 10 different scales ranging from 1 to 30, hence, 140 kernels for each dataset in total.

3.6.2 High-dimensional function inversion

In this section, we evaluate the ability of the different mapping methods to learn a smooth low- to high-dimensional function \mathbf{f} from noisy training examples in order to inverse it. In other words, given a new high-dimensional image $\mathbf{y} = \mathbf{f}(\mathbf{x})$, recover the low-dimensional \mathbf{x} . PLOM and PPAM were constrained with equal and isotropic covariance matrices $\{\Sigma_k\}_{k=1}^K$ as it showed to yield the best results. An equal number of components $K = 5$ was used in PLOM, PPAM and JGMM. Extensive experiments showed that obtained errors always decrease when K increases, although too high values of K lead to degenerate covariance matrices in classes where there are too few samples. Such classes are simply removed along the execution of the algorithms, thus reducing K . The choice of K was therefore not critical. For RVM, the kernel leading to the least average error in the interpolation of α out of 140 tested kernels was the *linear spline kernel* [Vapnik 97] with a scale parameter of 8. It was thus used for comparison.

Fully observed output We start by considering the case where $\mathbf{x} = \mathbf{t} \in \mathbb{R}^{L_t}$ is fully observed, *i.e.*, $L_w = 0$. We used a family of functions of the form $\mathbf{a} : [0, 10] \rightarrow \mathbb{R}^D$ ($L_t = 1$). Using the decomposition $\mathbf{a} = (a_1 \dots a_d \dots a_D)^\top$, each component a_d is defined

¹http://www.mvrvm.com/Multivariate_Relevance_Vector

by:

$$a_d(t) = \alpha_d \cos(\eta_d t/10 + \phi_d) \quad (3.36)$$

where $\{\alpha_d, \eta_d, \phi_d\}_{d=1}^D$ are scalars drawn uniformly at random from respectively $[0, 2]$, $[0, 4\pi]$ and $[0, 2\pi]$. This choice allows to generate a wide range of high-dimensional functions with different properties, *e.g.*, monotonicity, periodicity or sharpness. In particular, the generated functions are chosen to be rather challenging for the piecewise affine assumption made in PPAM and PLOM.

One hundred such functions were generated, and for each function, a set of N training couples $\{(t_n, \mathbf{y}_n)\}_{n=1}^N$ and a set of N' test couples $\{(t'_n, \mathbf{y}'_n)\}_{n=1}^{N'}$ were synthesized by uniformly drawing t values at random in the function's support intervals, and by adding some isotropic Gaussian noise e , *i.e.*, $\mathbf{y} = \mathbf{a}(t) + e$. Training couples were used to obtain a set of parameters with the PPAM and PLOM-EM algorithms. Then, the learned functions were inverted using the inverse mapping formula (3.12) to compute an estimate \hat{t}'_n given a test observation \mathbf{y}'_n .

Table 3.2 displays the average (Avg), standard deviation (Std) and percentage of *extreme values* (Ex) of the absolute errors $|\hat{t}'_n - t'_n|$ obtained with the different methods using for each generated function an observation dimension $D = 50$, an average signal to noise ratio (SNR²) of 3dB, $N' = 200$ training points and $N = 200$ test points, *i.e.*, 20,000 tests in total. We define *extreme values* (Ex) as those higher than the average error that would be obtained by an algorithm returning random values of t from the training set. This measure will be repeatedly used throughout this result section.

PPAM and PLOM perform significantly better than the 4 other methods in the task considered. PPAM (or equivalently PLOM-0) performed slightly better than PLOM-1 or PLOM-2. In other words, adding latent components did not improve the results for this task. This is the expected result, since the output t is fully observed during training in that case.

Partially latent output We now consider a situation where only some components of the function's support can be observed. The three function families used for testing are of the form $\mathbf{f} : [0, 10] \times [-1, 1] \rightarrow \mathbb{R}^D$, $\mathbf{g} : [0, 10] \times [0, 10] \rightarrow \mathbb{R}^D$ and $\mathbf{h} : [0, 10] \times [0, 10] \times [-1, 1] \rightarrow \mathbb{R}^D$. Each component is defined by:

$$f_d(t, w) = \alpha_d \cos(\eta_d t/10 + \phi_d) + \gamma_d w^3 \quad (3.37)$$

$$g_d(t, w) = \alpha_d \cos(\eta_d t/10 + \beta_d w + \phi_d) \quad (3.38)$$

$$h_d(t, w_1, w_2) = \alpha_d \cos(\eta_d t/10 + \beta_d w_1 + \phi_d) + \gamma_d w_2^3 \quad (3.39)$$

where $\mathbf{x} = [t; w]$ has an observed part t and a latent part w , and where $\{\alpha_d, \eta_d, \phi_d, \beta_d, \gamma_d\}_{d=1}^D$ are scalars drawn uniformly at random from respectively $[0, 2]$, $[0, 4\pi]$, $[0, 2\pi]$, $[0, \pi]$ and $[0, 2]$.

Again, one hundred functions of each of these three types were generated, and for each function a set of N training couples and a set of N' test couples were synthesized by

$${}^2\text{SNR} = 10 \log \frac{\|\mathbf{y}\|^2}{\|e\|^2}$$

Table 3.3: Average (Avg), standard deviation (Std) and percentage of extreme values (Ex) of the absolute error obtained with different methods for partially observed function f , g and h inversion and interpolation.

Method	f			g			h		
	Avg	Std	Ex	Avg	Std	Ex	Avg	Std	Ex
JGMM	2.20	2.44	24.4	2.37	2.47	26.6	2.74	3.19	29.7
SIR-1	1.58	1.29	12.3	1.43	1.14	6.87	1.66	1.31	12.7
SIR-2	0.73	0.80	1.95	0.89	0.86	2.60	1.10	1.06	4.63
RVM	0.70	0.55	0.05	0.85	0.67	0.52	1.00	0.80	1.28
PPAM	0.36	0.52	0.52	0.40	0.37	0.02	0.69	0.78	1.55
PLOM-1	0.24	0.24	0.00	0.34	0.34	0.00	0.56	0.59	0.49
PLOM-2	0.28	0.27	0.00	0.35	0.33	0.02	0.46	0.46	0.08
PLOM-3	0.29	0.29	0.02	0.36	0.38	0.06	0.47	0.47	0.16

uniformly drawing t and w values at random in the function’s support intervals, and by adding some isotropic Gaussian noise.

The same parameters as in the fully observed output case were used for each method, and models PLOM-1, PLOM-2 and PLOM-3 were tested. Results obtained using an observation dimension $D = 50$, $N = 200$ training points, $N' = 200$ test points (hence 20,000 tests per family of functions in total) and a signal to noise ratio (SNR³) of 3dB are showed in Table 3.3. The best results are always obtained when using PLOM- L_w^* where L_w^* is the actual dimension of the unobserved variable \mathbf{W} , demonstrating the effectiveness of the proposed partially-latent variable model. More than 30% improvement is measured with respect to the second best method PPAM (PLOM-0). If we compare results in Table 3.2 and Table 3.3, we logically observe that all the methods perform worse when some components of the outputs are missing. While PLOM-1 and PLOM-2 performed slightly worse than PPAM in the fully-observed case, they perform significantly better in this more challenging case, showing the advantage of modeling latent components of the output.

Figure 3.3(a) shows the influence of the observation space dimension D on the mean mapping error using various methods (average error for 20 synthesized functions h and 200 test points for each). While for low input dimension ($D < 10$) the 6 methods yield similar results, PPAM and PLOM-2 dramatically outperform the 4 others in higher dimension (mean PPAM error up to 39% lower than RVM). The addition of a two-dimensional latent component in PLOM decreases the error up to 39% with respect to PPAM for high values of D .

Figure 3.3(b) shows the influence of the signal-to-noise ratio (SNR) on the mean mapping error (average error for 20 synthesized functions f and 200 test points for each). Apart from JGMM which is very prone to overfitting due to its large number of parameters when D is high, all techniques perform similarly under extreme noise level (SNR = -10 dB). For higher SNRs, PPAM and PLOM significantly outperform the other techniques (mean PPAM error up to 45% lower than RVM). At high SNR (10dB) PLOM-1

³SNR = $10 \log \frac{\|\mathbf{y}\|^2}{\|\mathbf{e}\|^2}$

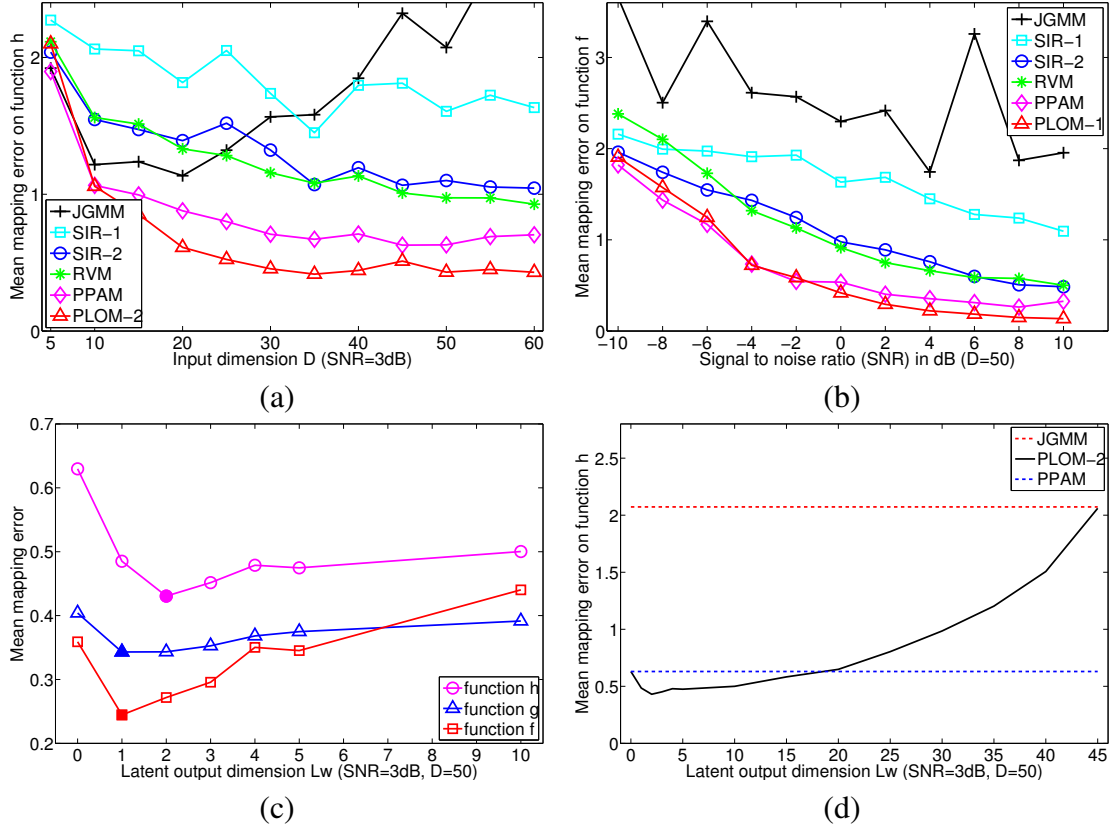


Figure 3.3: Influence of various parameters on the mean mapping error of synthetic functions using different regression techniques.

allows to decrease the error up to 59% compared to PPAM.

As illustrated in Figure 3.3(c), the best PLOM results are always obtained when the value chosen for the dimension of the latent component \mathbf{W} is the one used for synthesizing the data, *i.e.* $L_w = L_w^*$.

Finally, Figure 3.3(d) illustrates well how PLOM provides a whole range of alternative models *in between* PPAM and JGMM, as explained in section 3.4.2. Values of L_w in the range $1 \dots 20$ improve results upon PPAM which does not model unobserved variables. As L_w increases beyond L_w^* the number of parameters to estimate becomes larger and larger and the model becomes less and less constrained until becoming equivalent to JGMM (see section 3.4.2). This explains why PLOM's results are very close to those of JGMM when $L_w = D - 1$.

3.6.3 Robustly retrieving pose and light from face images

In this section, we test the different mapping methods on the *face dataset*⁴ which consists of 697 images (of size 64×64 pixels) of a 3D model of a head whose pose is parameterized

⁴<http://isomap.stanford.edu/datasets.html>



Figure 3.4: Example of face images from the Stanford’s face dataset.

by a left-right *pan* angle ranging from -75° to $+75^\circ$ and an up-down *tilt* angle ranging from -10° to $+10^\circ$. Example of such images are given in Figure 3.4. The image of a face depends on both the pose as well as on lighting that is absolutely necessary for rendering. The latter is simulated with one parameter taking integer values between 105 and 255. Images were down sampled to 16×16 and stacked into $D = 256$ dimensional vectors. In all the tasks considered, the algorithms were trained using a random subset of $N = 597$ images, and tested with the remaining $M = 100$ images. We repeated this train-then-test process 50 times for each task (5,000 tests per task in total). We used $K = 10$ for PPAM, PLOM and JGMM, but the same remarks as in section 3.6.2 apply. Again, PLOM and PPAM were constrained with equal and isotropic covariance matrices $\{\Sigma_k\}_{k=1}^K$ as it showed to yield the best results. Regarding RVM, as done previously, the best out of 140 kernels was used, *i.e.*, linear spline with scale 20.

Fully observed output The first task considered is to retrieve the 3-dimensional pose (2 angles) and light (1 value) information from a new input image. The algorithms are trained with images annotated with both pose and light parameters. Hence, the output is fully observed. Table 3.4 shows results obtained with the different methods. PPAM and PLOM outperforms the four other methods in terms of both pose and light estimation. Although all the output variables were fully-observed during training in this task, PLOM with 1 or 2 latent components perform slightly better than its fully-observed instance PPAM. While this improvement is not very large (around 8%), it shows that the more elaborated noise model induced by PLOM when $L_w > 0$ may improve results even for fully-observed-output mapping tasks.

Partially latent output We now consider two other tasks where the output is only partially annotated. Firstly the methods are used to learn the pose-to-image mapping using pairs of image-pose observations for training while the lighting is unobserved, *i.e.*, *light-invariant face pose estimation*. Secondly, the methods are used to learn the lighting-to-image mapping using pairs of image-light observations for training while the pose is unobserved, or *pose-invariant light-direction estimation*. Table 3.5 shows results obtained with the different methods. We show results obtained with PLOM- L_w^* and PLOM- L_w^\dagger . L_w^* denotes the ground-truth latent-component dimension, and L_w^\dagger is the latent-component

Table 3.4: Face dataset: Average (Avg), standard deviation (Std) and percentage of extreme values (Ex) of absolute pan and tilt angular errors and light errors obtained with different methods, when the 3 output variables are observed during training.

Method	Pan error ($^{\circ}$)			Tilt error ($^{\circ}$)			Light error		
	Avg	Std	Ex	Avg	Std	Ex	Avg	Std	Ex
JGMM	8.40	14.7	2.9	1.98	2.07	3.8	10.9	15.4	2.5
SIR-1	16.2	11.5	1.8	2.68	2.12	5.0	15.5	13.6	3.78
SIR-2	10.5	10.0	0.5	1.85	1.73	2.0	13.9	13.6	3.1
RVM	14.1	12.5	2.8	2.67	2.16	6.1	23.2	20.0	11
PPAM	4.29	4.68	0.0	1.67	1.46	1.1	7.46	6.68	0.0
PLOM-1	3.96	3.84	0.0	1.61	1.44	1.0	6.83	6.13	0.0
PLOM-2	3.94	4.02	0.0	1.56	1.36	0.7	6.89	6.08	0.0

Table 3.5: Face dataset: Average (Avg), standard deviation (Std) and percentage of extreme values (Ex) of (a) absolute pan and tilt angular errors when light is unobserved and (b) light errors when the pan and tilt angles are not observed. L_w^* is the true value of L_w while L_w^\dagger is the best found dimension in terms of empirical error.

Method	Pan error ($^{\circ}$)			Tilt error ($^{\circ}$)			Light error		
	Avg	Std	Ex	Avg	Std	Ex	Avg	Std	Ex
JGMM	13.2	26.6	8.2	2.32	3.01	7.0	18.2	21.0	6.7
SIR-1	16.0	11.3	1.4	2.64	2.06	4.9	15.2	13.2	3.2
SIR-2	10.6	9.73	0.4	1.81	1.66	1.9	13.6	13.2	2.8
RVM	14.0	12.2	1.9	2.63	2.13	5.8	18.7	15.7	4.82
PPAM	6.06	5.49	0.0	1.79	1.58	1.4	10.7	8.94	0.1
PLOM- L_w^*	3.78	4.11	0.0	1.61	1.46	1.0	10.5	9.14	0.3
PLOM-L_w^\dagger	2.76	3.11	0.0	1.17	1.13	0.4	8.78	7.79	0.1

(a)

(b)

dimension which empirically showed the best results, when varying L_w between 0 and $L_w^{\max} = D = 256$. For light-invariant face pose estimation the ground truth latent-component dimension is $L_w^* = 1$, and we obtained the best results with $L_w^\dagger = 12$. For pose-invariant light-direction estimation the ground truth latent-component dimension is $L_w^* = 2$, and we obtained the best results with $L_w^\dagger = 13$. Overall, PLOM- L_w^\dagger achieved a 20% to 60% improvement with respect to the second best method PPAM. As expected, the improvement of adding latent components to the PLOM model is much more significant in this partially-latent-output mapping problem than it was for the fully-observed-output mapping problem.

Based on these experiments, an interesting observation is that, although the ground-truth dimension L_w always reduces the mean error with respect to $L_w = 0$ (PPAM), the error is further reduced by selecting a latent dimension larger than the true one. This suggests that the actual local linear effect of the latent variable \mathbf{W} on the observed variable \mathbf{Y} could be modeled more accurately by choosing a latent dimension that is higher than the “expected” dimension.













































Input image	PLOM estimates	Reconstructions for different values of w ($L_w = 1$)									Rec. PPAM
		$w=-2$	-1.5	-1	-0.5	0	+0.5	+1	+1.5	+2	
	$t_1 = -41^\circ$ $t_2 = 8.7^\circ$ $w = 1.73$										
	$t_1 = 55^\circ$ $t_2 = -5.4^\circ$ $w = 0.28$										
	$t_1 = -9.8^\circ$ $t_2 = 4.3^\circ$ $w = -1.47$										
	$t_1 = -24^\circ$ $t_2 = 8.2^\circ$ $w = 1.32$										

Figure 3.5: Recovering the 3D pose of a face (t_1 =pan angle, t_2 =tilt angle) with lighting being modeled by the latent variable W . (a) The input image. (b) The pose and lighting estimates using PLOM. (c) Reconstructed images using the estimated pose parameters and different values for w . (d) Reconstructed images using the pose parameters estimated using PPAM.

Another experiment was run to verify whether the latent variable values recovered with our method were meaningful. Once a set of model parameters $\tilde{\theta}$ were estimated using PLOM-1 and with a training set of 597 pose-to-image associations, a new test image \mathbf{y} was selected at random and was used to recover both $\hat{\mathbf{t}} \in \mathbb{R}^2$ and $\hat{w} \in \mathbb{R}$ based on the inverse conditional expectation $\hat{\mathbf{x}} = [\hat{\mathbf{t}}; \hat{w}] = \mathbb{E}[\mathbf{X}|\mathbf{y}; \tilde{\theta}]$ (3.12). An image was then reconstructed using the forward conditional expectation $\hat{\mathbf{y}} = \mathbb{E}[\mathbf{Y} | [\hat{\mathbf{t}}; \hat{w}]; \tilde{\theta}]$ (3.11) while varying the value of w in order to visually observe its influence on the reconstructed image. Results obtained for different test images are displayed in Fig. 3.5. These results show that the latent variable W of PLOM does capture lighting effects, whereas an explicit lighting parameterization was not present in the training set. For comparison, we show images obtained after projection and reconstruction using PPAM. As it may be observed, the image reconstructed with PPAM looks like a blurred average over all possible lightings, while PLOM allows a much more accurate image reconstruction process. This is because PLOM encodes images with 3 rather than 2 variables, one of which being latent and estimated in an unsupervised way.

3.6.4 Retrieval of Mars physical properties from hyperspectral images

Visible and near infrared imaging spectroscopy is a key remote sensing technique used to study and monitor planets. It records the visible and infrared light reflected from the planet in a given wavelength range and produces cubes of data where each observed surface location is associated with a spectrum. Physical properties of the planets' surface, such as chemical composition, granularity, texture, etc, are some of the most important param-

ters that characterize the morphology of spectra. In the case of Mars, radiative transfer models have been developed to numerically evaluate the link between these parameters and observable spectra. Such models allow to simulate spectra from a given set of parameter values, *e.g.*, [Douté 07]. In practice, the goal is to scan the Mars ground from an orbit in order to observe gas and dust in the atmosphere and look for signs of specific materials such as silicates, carbonates and ice at the surface. We are thus interested in solving the associate inverse problem which is to deduce physical parameter values from the observed spectra. Since this inverse problem cannot generally be solved analytically, the use of optimization or statistical methods has been investigated, *e.g.* [Bernard-Michel 09]. In particular, training approaches have been considered with the advantage that, once a relationship between parameters and spectra has been established through training, the learned relationship can be used for very large datasets and for all new images having the same physical model.

Within this category of methods, we investigate in this section the potential of the proposed PPAM and PLOM models using a dataset of hyperspectral images collected from the imaging spectrometer OMEGA instrument [Bibring 04] on-board of the Mars express spacecraft. To this end a database of synthetic spectra with their associated parameter values were generated using a radiative transfer model. This database is composed of 15,407 spectra associated with five real parameter values, namely, proportion of water ice, proportion of CO₂ ice, proportion of dust, grain size of water ice, and grain size of CO₂ ice. Each spectrum is made of 184 wavelengths. A mapping method can be used to learn a relationship between parameters and spectra from the synthetic database, and then to estimate the corresponding parameters of a new real-world spectrum. Since no ground truth is available for Mars, the synthetic database also served as a first test set to evaluate the accuracy of the predicted parameter values. An objective evaluation was done by cross validation: For all methods, we selected 10,000 training couples at random from the synthetic set, tested on the 5,407 remaining spectra, and repeated this 20 times.

We used $K = 50$ for PPAM, PLOM and JGMM. PPAM and PLOM were constrained with equal, diagonal covariance matrices as it showed to yield the best results. For all algorithms, training data were normalized to have 0 mean and unit variance using scaling and translating factors. These factors were then used on test data and estimated output to obtain final estimates. This technique showed to noticeably improve results of all methods. As regards RVM, the best out of 140 kernels was used: a third degree polynomial kernel with scale 6 showed the best results using cross-validation on the database. As a quality measure of the estimated parameters, we computed normalized root mean squared errors (NRMSE⁵). The NRMSE quantifies the difference between the estimated and real parameter values. This measure is normalized enabling direct comparison between the parameters which are of very different range. The closer NRMSE is to zero the more accurate are the predicted values.

$${}^5\text{NRMSE} = \sqrt{\frac{\sum_{m=1}^M (\hat{t}_m - t_m)^2}{\sum_{m=1}^M (t_m - \bar{t})^2}} \text{ with } \bar{t} = M^{-1} \sum_{m=1}^M t_m.$$

Table 3.6: Normalized root mean squared error (NRMSE) for Mars surface physical properties recovered from hyperspectral images, using synthetic data, different methods and fully-observed-output training.

Method	Prop. H ₂ O	Prop. CO ₂	Prop. Dust	Size H ₂ O	Size CO ₂
JGMM	2.40 ± 18.5	0.84 ± 1.64	0.63 ± 1.02	0.73 ± 1.02	1.08 ± 4.52
SIR-1	3.41 ± 20.0	1.28 ± 2.16	1.04 ± 1.79	0.69 ± 0.92	1.85 ± 7.24
SIR-2	3.27 ± 18.6	0.96 ± 1.75	0.89 ± 1.53	0.62 ± 0.86	1.66 ± 6.53
RVM	1.28 ± 7.57	0.50 ± 0.95	0.40 ± 0.69	0.51 ± 0.67	0.89 ± 3.80
PPAM	1.04 ± 6.66	0.37 ± 0.72	0.28 ± 0.50	0.45 ± 0.74	0.60 ± 2.59
PLOM-1	0.95 ± 5.92	0.34 ± 0.65	0.24 ± 0.44	0.42 ± 0.71	0.56 ± 2.44
PLOM-2	0.99 ± 6.02	0.36 ± 0.70	0.27 ± 0.48	0.40 ± 0.66	0.58 ± 2.66

Table 3.7: Normalized root mean squared error (NRMSE) for Mars surface physical properties recovered from hyperspectral images, using synthetic data, different methods and partially-latent-output training.

Method	Proportion of CO ₂ ice	Proportion of dust	Grain size of water ice
JGMM	0.83 ± 1.61	0.62 ± 1.00	0.79 ± 1.09
SIR-1	1.27 ± 2.09	1.03 ± 1.71	0.70 ± 0.94
SIR-2	0.96 ± 1.72	0.87 ± 1.45	0.63 ± 0.88
RVM	0.52 ± 0.99	0.40 ± 0.64	0.48 ± 0.64
PPAM	0.54 ± 1.00	0.42 ± 0.70	0.61 ± 0.92
PLOM-1	0.36 ± 0.70	0.28 ± 0.49	0.45 ± 0.75
PLOM-2*†	0.34 ± 0.63	0.25 ± 0.44	0.39 ± 0.71
PLOM-3	0.35 ± 0.66	0.25 ± 0.44	0.39 ± 0.66
PLOM-4	0.38 ± 0.71	0.28 ± 0.49	0.38 ± 0.65
PLOM-5	0.43 ± 0.81	0.32 ± 0.56	0.41 ± 0.67
PLOM-20	0.51 ± 0.94	0.38 ± 0.65	0.47 ± 0.71

Fully observed output We start by retrieving the 5 parameter values, namely, proportion of water ice (Prop. H₂O), proportion of CO₂ ice (Prop. CO₂), proportion of dust (Prop. Dust), grain size of water ice (Size H₂O), and grain size of CO₂ ice (Size CO₂) from 184-dimensional spectra using the different mapping methods. The training was done with synthetic spectra annotated with the 5 parameters, and hence output variables were fully observed. Results obtained with the 6 methods are showed in table 3.6. PPAM and PLOM performed similarly and outperform the 4 other methods in estimating each parameter. As expected, using 1 or 2 additional latent components in PLOM did not show any significant improvement compared to PPAM in this task, since the output is fully observed during training. Notice that obtained mean NRMSE with the proportion of water (column 2) and the grain size of CO₂ (column 6) parameters are very high, *i.e.*, more than 0.5 for all methods. This suggest that the relationship between these parameters and observed spectra is complex and harder to learn.

Partially latent output In order to fully illustrate the potential of PLOM, we now deliberately ignore two of the parameters in the database and consider them as latent variables.

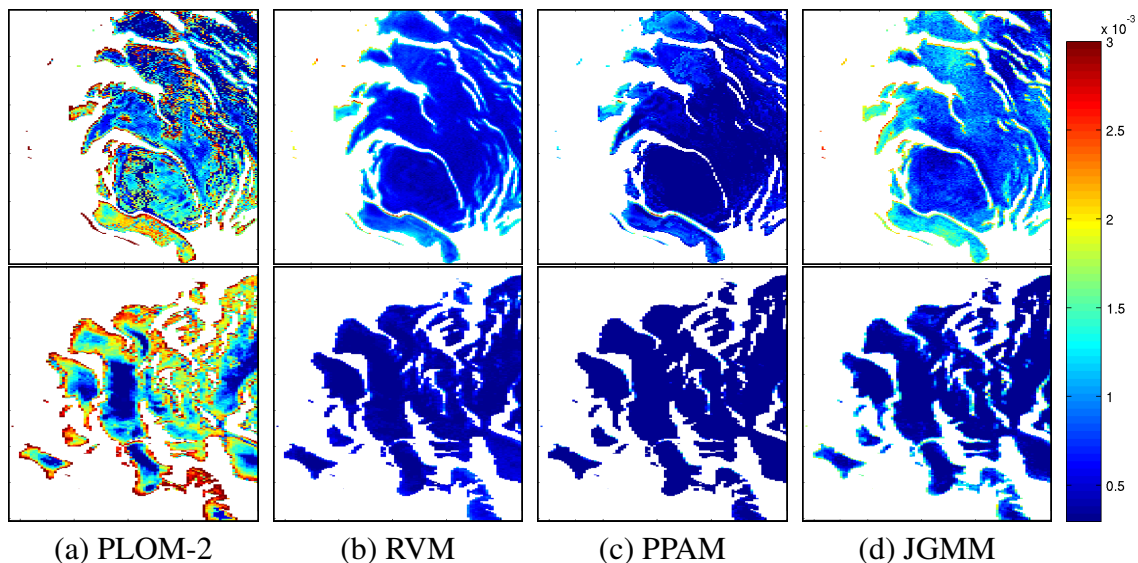


Figure 3.6: Proportion of dust obtained with 4 different mapping methods on real data. The data correspond to hyperspectral images grabbed from two different viewpoints of the South polar cap of Mars. First row: orbit 41, second row: orbit 61. White areas correspond to unexamined regions, where the synthetic model does not apply.

We chose to ignore the proportion of water ice and the grain size of CO_2 ice, as these are the ones yielding the poorest reconstruction error when outputs are fully observed (see Table 3.6). In addition, we observed that using them in the inversion tend to degrade the estimation of the other three parameters, which are of particular interest, namely proportion of CO_2 ice, proportion of dust and grain size of water ice. These two parameters appear in some previous study [Bernard-Michel 09] to be sensitive to the same wavelengths than the proportion of dust and are suspected to mix with the other parameters in the synthetic transfer model so that they are harder to estimate. Therefore, we excluded them, treated them as latent variables, and did the regression with the three remaining parameters.

Table 3.7 shows obtained NRMSE for the three parameters considered. The ground truth latent variable dimension is $L_w^* = 2$, and accordingly, the empirically best dimension for PLOM was $L_w^\dagger = 2$. PLOM-2 outperformed all the other methods on that task, with 36% with the second best method RVM, closely followed by PPAM. Note that the computational and memory costs of RVM for training were one order of magnitude higher than those of PLOM, using Matlab implementations. Interestingly, notice how for almost all methods, removing the “faulty” parameters that were harder to estimate in the fully-observed training slightly decreased the reconstruction error of the remaining three others, as compared to Table 3.6.

We then used the synthetic database to train the algorithms and test them on real data made of observed spectra. In particular, we focus on a dataset of Mars’s South polar cap. Since no ground truth is currently available for the physical properties of Mars polar regions, we propose a qualitative evaluation. We used the 4 best methods among the tested

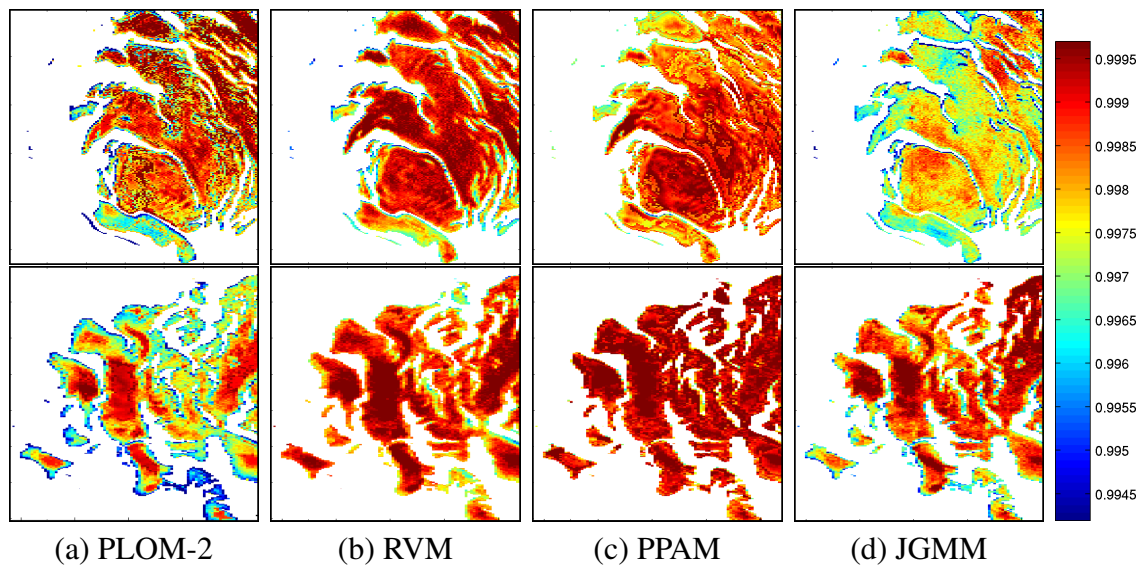


Figure 3.7: Proportion of CO₂ ice obtained from hyperspectral images of two different viewpoints of the South polar cap of Mars. First row: orbit 41, second row: orbit 61. White areas correspond to unexamined regions, where the synthetic model does not apply.

ones, namely PLOM-2, RVM, PPAM and JGMM, to retrieve the physical properties of the South polar cap using two hyperspectral images of approximately the same area from different view points (orbit 41 and orbit 61). Since we are looking for proportions between 0 and 1, returned values smaller than 0 or higher than 1 are not acceptable and hence they were set to one of the bounds. As it can be seen in Fig. 3.6 and Fig. 3.7, PLOM outputs proportion maps with similar characteristics for the two view points, which suggests good consistency. Such a consistency is not observed using the other tested methods. In addition, RVM and PPAM gave a higher number of values falling outside the interval $[0, 1]$. Moreover, PLOM-2 is the only method featuring less dust at the South pole cap center and higher concentrations of dust at the boundaries of the CO₂ ice, which matches expected results from planetology [Douté 05]. Finally, note that the proportions of CO₂ ice and dust clearly seem to be complementary using PLOM-2, while this complementarity is less obvious using other methods.

3.6.5 2D localization of a white noise sound source

We finally compare the mapping methods on one of the central problems addressed in this thesis, namely sound source localization. Note that all the mapping methods considered in this section perform a mapping from a vector-valued input to a vector-valued output. However, typical audio inputs obtained from the binaural recording of sound sources have the shape of a spectrogram, *i.e.*, a noisy time series of interaural feature vectors, possibly mixed, and possibly with missing values (see section 2.2.2). Extending GLLiM models to deal with such inputs will be the focus of chapters 4 and 5 of this thesis. However, as detailed in section 2.2.2, an interaural spectrograms obtained from the binaural recording

Table 3.8: White-noise sound source localization using the audio-visual dataset (Section 2.3.3). The table shows average (Avg), standard deviation (Std) and percentage of extreme values (Ex) of absolute errors on horizontal and vertical axis in pixels, obtained with different methods.

Method	Horizontal axis			Vertical axis		
	Avg	Std	Ex	Avg	Std	Ex
JGMM	18.0	18.4	0.0	26.4	27.4	0.4
SIR-1	58.3	52.1	1.2	71.2	52.8	6.6
SIR-2	42.0	38.1	0.3	56.5	47.4	3.6
RVM	35.1	27.9	0.0	65.1	50.1	5.2
PPAM	14.0	12.3	0.0	26.2	23.1	0.1
PLOM-1	14.2	13.2	0.0	24.6	21.8	0.0
PLOM-2	13.7	12.5	0.0	24.4	22.0	0.1
PLOM-4	13.3	12.5	0.0	23.2	21.5	0.1
PLOM-9[†]	12.7	12.3	0.0	22.4	21.4	0.0

of a single, static white noise emitter does not contain missing value and can thus be averaged to obtain a less noisy mean interaural feature vector. These high-dimensional vectors were showed to lie on a smooth, locally-linear, L -dimensional manifold parameterized by the sound source direction in section 2.4. Mapping interaural feature vectors to sound source directions is thus exactly in the scope of the proposed PPAM and PLOM models, and can also be performed by JGMM, SIR and RVM.

We used $K = 30$ for PPAM, PLOM and JGMM. Noise covariance matrices of PPAM and PLOM were constrained to be equal and diagonal. Based on cross-validation on the data, we chose the best out of 140 kernels for RVM. The one with lowest localization errors was the *thin-plate spline kernel* [Wahba 90] with a scale parameter of 6. We used concatenation of ILD feature vectors (512 dimensions) and IPD vectors (1,024 dimension) to obtain $D = 1,536$ -dimensional ILPD feature vectors. The dimension of IPD vectors is twice the dimension of ILD vectors because they are expressed in \mathbb{C} (or equivalently \mathbb{R}^2) instead of $]-\pi, \pi]$ via (2.4). This is because none of the regression methods considered can deal with circular values.

Fully observed output We first used the white noise recordings of the audio-visual datasets presented in section 2.3.3 for training and testing. The algorithms were trained using $N = 232$ randomly picked interaural-feature-vectors annotated with their corresponding source position in the image, in pixels. They were then tested by estimating the source position of the remaining $N' = 200$ interaural feature vectors. This was repeated 20 times, for a total of 4,000 white noise sound source localization tests. Localization errors in pixels on the horizontal and vertical axis for different methods are given in table 3.8. Recall that the camera used in the setup has 480×640 and $\approx 21^\circ \times 28^\circ$ field of view. Assuming a one-to-one mapping from pixel coordinates to 2D directions, it follows that 30 pixels span $\approx 1.3^\circ$, or 3.5cm at the experimental range. As can be seen, PPAM and PLOM provide a very high localization accuracy (less than 1° horizontally and vertically)

Table 3.9: White-noise sound source localization using the cluttered audio-visual dataset. The table shows average (Avg), standard deviation (Std) and percentage of extreme values (Ex) of absolute errors on horizontal and vertical axis in pixels, obtained with different methods.

Method	Horizontal axis			Vertical axis		
	Avg	Std	Ex	Avg	Std	Ex
JGMM	51.7	59.5	3.1	74.3	78.1	13
SIR-1	66.8	57.9	3.2	94.4	68.9	19
SIR-2	54.9	50.4	1.6	85.6	66.2	14
RVM	55.2	46.6	1.0	97.1	66.1	20
PPAM	20.8	15.9	0.0	56.3	50.8	4.0
PLOM-4*	20.4	19.6	0.1	49.9	43.9	2.4
PLOM-9†	20.5	19.6	0.1	46.3	38.8	1.4

and significantly outperform the 4 other mapping methods. The second best method using these data is JGMM. However, as will be seen later, JGMM’s good results are probably due to overfitting, because of the low amount of noise in considered data.

Only some slight improvement with respect to PPAM is obtained by adding a few latent component to PLOM. The latent component dimension yielding the best results between 1 and 10 was $L_w = 9$, decreasing the horizontal error of 11% and the vertical error of 8% with respect to PPAM. This relatively small improvement suggests that interaural feature vectors mostly depends on the 2D source position in this dataset, and are barely perturbed by the effects of other latent variables.

Partially latent output In order to put forward the advantage of PLOM for sound source localization, we ran preliminary experiments using a new specifically built dataset. This dataset was built with the audio-visual acoustic space sampling method detailed in section 2.3.3, and will be referred to as the *cluttered* audio-visual dataset. In this dataset, some form of *slack* is allowed in the spatial characteristics of the emitter, in order to see if that slack can be captured by the latent variable of PLOM. In the standard audio-visual dataset the emitter is approximately kept at the same distance and oriented towards the center of the dummy head. In the cluttered audio-visual dataset, we deliberately performed some random manual rotation of the emitter around its center, and varied the listener-to-emitter distance at each position. The maximum variations in the emitter’s distance were in the order of a meter. The maximum variations in the emitter’s orientation were in the order of 45° in azimuth and elevation. Such variations should have impact on interaural feature vectors due to the complex reverberating properties of the recording room. Since the emitter self orientation and distance cannot be easily annotated, they are unobserved during training, and should be captured by the latent part of the output in the PLOM model. A dataset containing 432 source positions on a 18×24 regular grid covering the image was recorded. A random subset of 232 points was used for training the algorithms, and the remaining 200 points were used for testing. This was repeated 20 times, for a total of 4,000 white noise sound source localization tests. Results obtained with different methods are showed in table 3.9. We show results obtained with PLOM- L_w^*

and PLOM– L_w^\dagger , where L_w^* is the ground-truth latent-component dimension, and L_w^\dagger is the latent-component dimension which empirically showed the best results between 1 and 10. For this task the ground-truth latent-component dimension is considered to be 4 (the distance and the three orientation angles of the emitter). Best results were obtained using $L_w^\dagger = 9$. As expected, all the regression algorithms considered perform worse on this more challenging dataset. The most dramatic decrease of performance occurred with JGMM, with localization errors 3 times larger. This is probably because JGMM tends to overfit the data, which explains why it performed well on the non-cluttered – and hence less noisy – dataset. Surprisingly, adding latent components to PLOM did not improve the horizontal localization accuracy. However, a significant 18% improvement is observed in vertical accuracy using the best model PLOM-9. These preliminary results encouragingly suggest that PLOM with positive values of L_w could be used to address more robustly mapping-based sound source localization in real world environment. More thorough experiments including notably changes in the room properties should be ran to further assess this idea.

3.7 Conclusion on Probabilistic Space Mapping

In this chapter, we explored, devised and unified a number of models for high-to-low dimensional regression in a probabilistic framework. In the four datasets considered, the proposed PPAM and PLOM methods significantly outperformed three other existing techniques, namely RVM, JGMM and SIR. PLOM showed to be particularly advantageous in situations where the output variable is partially annotated. Note that the best kernel choice for RVM was different in all four datasets. In fact, very large differences in performance were observed depending on the kernel used and the data considered. Choosing an appropriate kernel type and scale for a given dataset cannot be done automatically and is a long and fastidious task. This constitutes a major drawback of RVM, and more generally of all kernel methods for regression, *e.g.* [Smola 04], [Lawrence 05], [Wu 08]. In contrast, PLOM only requires the choice of two integer parameters K and L_w , both having an intuitive interpretation. K represents the number of approximately affine components in the mapping function to estimate, and L_w represents the number of latent variable affecting the observed data. All experiments showed that the choice of K was not critical, since larger values usually leads to lower errors, while too high value automatically decreases the number of component by removing empty clusters. An open topic for future research is how to automatically estimate K and L_w . The generative nature of PLOM may allow to treat this issue as a model selection problem and to consider standard information criteria, such as the Bayesian information criterion, or to adapt techniques for estimating the intrinsic dimension in high dimensional data [Bouveyron 11].

Although the last experiments of section 3.6.5 encouragingly suggests that adding latent component to PLOM could improve sound source localization robustness in challenging scenarios, this improvement was not significant using our non-cluttered audio-visual dataset. The relatively small improvement obtained is at the cost of an increased complexity of the model, and more computational time. Therefore, the more simple PPAM

model will be used for the sound sources localization tasks addressed in the remainder of this thesis. Note, however, that all the extensions developed for PPAM are general, and directly applicable to the PLOM models considered in this chapter.

Appendix

3.A Proof of Theorem 1

This appendix provides a proof of theorem 1, *i.e.*, an unconstrained Gaussian locally linear mapping (GLLiM) model is equivalent to an unconstrained joint Gaussian mixture model (JGMM). Conversion formulas (3.6) are obtained using (3.7) and formulas for conditional multivariate Gaussian variables. Conversion formulas (3.7) are obtained from standard algebra by identifying the joint distribution $p(\mathbf{X}, \mathbf{Y} | \mathbf{Z}; \boldsymbol{\theta})$ defined by (3.2) and (3.3) with a multivariate Gaussian distribution.

To complete the proof, one need to prove the following two statements:

(i) For any $\rho_k \in \mathbb{R}$, $\mathbf{m}_k \in \mathbb{R}^{D+L}$ and $\mathbf{V}_k \in \mathcal{S}_+^{L+D}$, there is a set of parameters $\mathbf{c}_k \in \mathbb{R}^L$, $\boldsymbol{\Gamma}_k \in \mathcal{S}_+^L$, $\pi_k \in \mathbb{R}$, $\mathbf{A}_k \in \mathbb{R}^{D \times L}$, $\mathbf{b}_k \in \mathbb{R}^D$ and $\boldsymbol{\Sigma}_k \in \mathcal{S}_+^D$ such that (3.6) holds.

(ii) Reciprocally, for any $\mathbf{c}_k \in \mathbb{R}^L$, $\boldsymbol{\Gamma}_k \in \mathcal{S}_+^L$, $\pi_k \in \mathbb{R}$, $\mathbf{A}_k \in \mathbb{R}^{D \times L}$, $\mathbf{b}_k \in \mathbb{R}^D$, $\boldsymbol{\Sigma}_k \in \mathcal{S}_+^D$ there is a set of parameters $\rho_k \in \mathbb{R}$, $\mathbf{m}_k \in \mathbb{R}^{L+D}$ and $\mathbf{V}_k \in \mathcal{S}_+^{D+L}$ such that (3.7) holds.

Where \mathcal{S}_+^M denotes the set of $M \times M$ symmetric positive definite matrices. We introduce the following lemma:

Lemma 1 If $\mathbf{V} = \begin{bmatrix} \mathbf{V}^{xx} & \mathbf{V}^{xy} \\ \mathbf{V}^{xy\top} & \mathbf{V}^{yy} \end{bmatrix} \in \mathcal{S}_+^{L+D}$, then $\boldsymbol{\Sigma} = \mathbf{V}^{yy} - \mathbf{V}^{xy\top} \mathbf{V}^{xx-1} \mathbf{V}^{xy} \in \mathcal{S}_+^D$.

Proof: Since $\mathbf{V} \in \mathcal{S}_+^{L+D}$ we have $\mathbf{u}^\top \mathbf{V} \mathbf{u} > 0$ for all non null $\mathbf{u} \in \mathbb{R}^{L+D*}$. Using the decomposition $\mathbf{u} = [\mathbf{u}^x; \mathbf{u}^y]$ we obtain

$$\mathbf{u}^{x\top} \mathbf{V}^{xx} \mathbf{u}^x + 2\mathbf{u}^{x\top} \mathbf{V}^{xy} \mathbf{u}^y + \mathbf{u}^{y\top} \mathbf{V}^{yy} \mathbf{u}^y > 0 \quad \forall \mathbf{u}^x \in \mathbb{R}^{L*}, \forall \mathbf{u}^y \in \mathbb{R}^{D*}.$$

In particular, for $\mathbf{u}^x = -\mathbf{V}^{xx-1} \mathbf{u}^y \mathbf{V}^{xy}$ we obtain

$$\mathbf{u}^{y\top} (\mathbf{V}^{yy} - \mathbf{V}^{xy\top} \mathbf{V}^{xx-1} \mathbf{V}^{xy}) \mathbf{u}^y > 0 \Leftrightarrow \mathbf{u}^{y\top} \boldsymbol{\Sigma} \mathbf{u}^y > 0 \quad \forall \mathbf{u}^y \in \mathbb{R}^{D*}$$

and hence $\boldsymbol{\Sigma} \in \mathcal{S}_+^D$ ■.

Lemma 2 If $\mathbf{A} \in \mathbb{R}^{D \times L}$, $\boldsymbol{\Gamma} \in \mathcal{S}_+^L$, $\boldsymbol{\Sigma} \in \mathcal{S}_+^D$, then $\mathbf{V} = \begin{bmatrix} \boldsymbol{\Gamma} & \boldsymbol{\Gamma} \mathbf{A}^\top \\ \mathbf{A} \boldsymbol{\Gamma} & \boldsymbol{\Sigma} + \mathbf{A} \boldsymbol{\Gamma} \mathbf{A}^\top \end{bmatrix} \in \mathcal{S}_+^{L+D}$.

Proof: Since $\boldsymbol{\Gamma} \in \mathcal{S}_+^L$ there is a unique symmetric positive definite matrix $\boldsymbol{\Lambda} \in \mathcal{S}_+^L$ such that $\boldsymbol{\Gamma} = \boldsymbol{\Lambda}^2$. Using standard algebra, we obtain that for all non null $\mathbf{u} = [\mathbf{u}^x; \mathbf{u}^y] \in \mathbb{R}^{L+D*}$,

$$\mathbf{u}^\top \mathbf{V} \mathbf{u} = \|\boldsymbol{\Lambda} \mathbf{u}^x + \boldsymbol{\Lambda} \mathbf{A}^\top \mathbf{u}^y\|^2 + \mathbf{u}^{y\top} \boldsymbol{\Sigma} \mathbf{u}^y$$

where $\|\cdot\|$ denotes the standard Euclidean distance. The first term of the sum is positive for all $[\mathbf{u}^x; \mathbf{u}^y] \in \mathbb{R}^{L+D^*}$ and the second term strictly positive for all $\mathbf{u}^y \in \mathbb{R}^{D^*}$ because $\Sigma \in \mathcal{S}_+^D$ by hypothesis. Therefore, $\mathbf{V} \in \mathcal{S}_+^{L+D}$ ■.

Lemma 1 and the correspondence formula (3.6) proves **(i)**, Lemma 2 and the correspondence formula (3.7) proves **(ii)**. This completes the proof of Theorem 1 ■.

CHAPTER 4

MAPPING-BASED SOUND SOURCE LOCALIZATION

We now address the problem of localizing a single source emitting *natural sounds* such as speech based on binaural recordings. This long-studied problem is here addressed in a *supervised* framework, *i.e.*, using the white-noise audio-motor or audio-visual training sets presented in Section 2.3. As explained in Section 2.2, the binaural recording of a white-noise emitter allow to obtain an interaural feature vector by taking the temporal mean of corresponding ILD and IPD spectrograms. Since these vectors lie on a smooth manifold parameterized by the source direction, localizing a *white-noise sound* is straightforward, either using a space mapping method as done in section 3.6.5, or by exhaustive nearest-neighbor search in the training set. The major difficulty in localizing *natural sounds* such as speech is that interaural spectrogram inputs consist in noisy times series of interaural vectors containing a lot of irrelevant or *missing* interaural cues, because the source is not emitting in all time-frequency points. This issue is formalized and detailed in section 4.1. We then propose two different approaches to map an interaural spectrogram to a source position. In section 4.2, a piecewise-constant approximation of the binaural manifold is considered, leading to a technique referred to as *probabilistic piecewise-constant mapping* (PPCM). PPCM maybe view as a probabilistic extension of nearest-neighbor to the case of spectrogram inputs. In section 4.3, we generalize the *probabilistic piecewise-affine mapping* (PPAM) technique presented in section 3.3 to spectrogram inputs, based on Bayes rules and standard algebra. We thoroughly evaluate the two methods using speech recordings from the audio-motor and the audio-visual datasets, and show their superiority in sound localization accuracy compared to a baseline method. We also test the most efficient method on a realistic sound source localization scenario involving a human speaker in real-world, reverberant conditions.

4.1 Sparsity of Natural Sound Spectrograms

In section 2.3, we explained how to record large white noise datasets allowing to associate 2D source positions $\{\mathbf{x}_n\}_{n=1}^N \subset \mathcal{X} \subset \mathbb{R}^L$ ($L = 2$) and recorded mean interaural features $\{\mathbf{y}_n\}_{n=1}^N \subset \mathcal{Y} \subset \mathbb{R}^D$. We denote $\mathcal{T} = \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^N$ such a set of associations. We would like to use \mathcal{T} to localize a new sound, given a binaural recording. As detailed in section 2.2, binaural recordings allow to obtain time series of interaural feature vectors (ILD, IPD or a concatenation of both). Let's denote such a time series $\{\mathbf{y}'_1, \dots, \mathbf{y}'_T\} \subset \mathbb{R}^D$. In the case of a white-noise emitter, all interaural feature values $\{y'_{dt}\}_{d=1, t=1}^{D, T}$ are present. In this chapter, we address the more complex case of a natural sound emitter such as speech. Since the source does not emit in all time-frequency points, several vectors of this series will have missing values. To characterize these values, we introduce the binary variables χ_{dt} so that $\chi_{dt} = 0$ if the value y'_{dt} is missing and $\chi_{dt} = 1$ otherwise. We note $\boldsymbol{\chi} = \{\chi_{dt}\}_{d, t=1}^{D, T}$. One way to determine such missing values is to use a threshold on the recorded total spectral power density (TSPD) $10 \log_{10}(|s_{ft}^{(L)}|^2 + |s_{ft}^{(R)}|^2)$. In practice, we manually set this threshold to -20dB based on the average background-noise spectral density. This is illustrated in Figure 4.1.

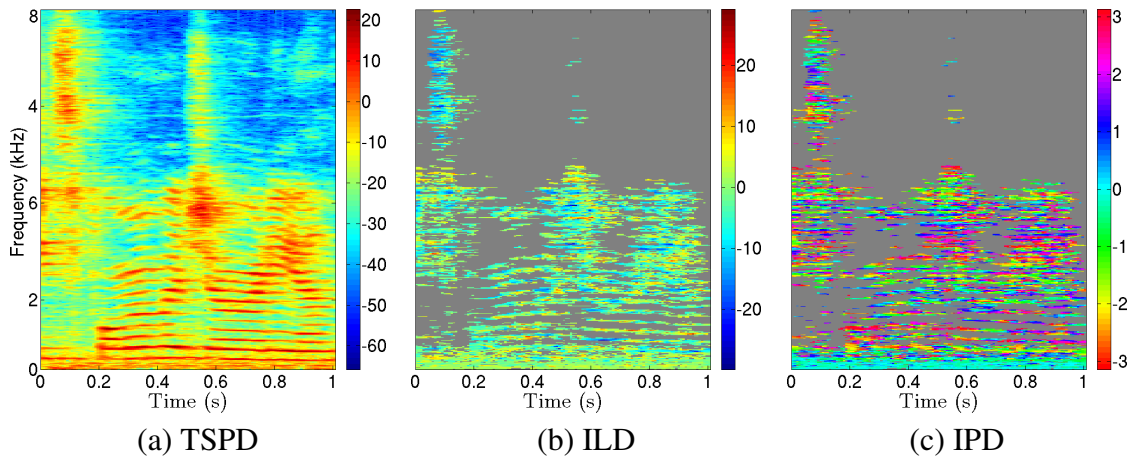


Figure 4.1: Spectrograms obtained from the 1 second binaural recording of a single speech source. Gray colors denote missing values. The threshold used on TSPD to determine missing values is -20dB .

Note that in all the sound localization tasks addressed in this thesis, the emitting sound source will be considered static along the T recorded spectrogram windows. Thus, the T binaural feature vectors are *redundant*, in that they capture the same spatial information. Due to the microphones, the background noise and the properties of discrete Fourier transform, these feature vectors are also very noisy. In summary, an interaural spectrogram input consists in a noisy, redundant time series of interaural feature vectors with missing values, denoted by $\mathcal{S} = \{\{\mathbf{y}'_1, \dots, \mathbf{y}'_T\}, \boldsymbol{\chi}\}$. Examples of such inputs are given in Figure 4.1(b) and 4.1(c).

4.2 Piecewise Constant Mapping

Given a training set $\mathcal{T} = \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^N$ and a spectrogram input \mathcal{S} , how can we estimate the position \mathbf{x} of the emitting sound source? We start by investigating methods that always return a sound source position \mathbf{x}_n in the training set \mathcal{T} . In the case of a single, complete, vector value input $\mathbf{y}'_t \in \mathbb{R}^D$, this would amount to find the “nearest” neighbor $\mathbf{y}_{\tilde{n}}$ of \mathbf{y}'_t in $\{\mathbf{y}_n\}_{n=1}^N$ and return $\mathbf{x}_{\tilde{n}}$, where “nearest” needs to be defined. We study in this section the more general case of spectrogram inputs. This can be formalized by the minimization of a discrete cost function $C_{\mathcal{S}}$ that takes values in $[1 : N]$ and depend on the spectrogram input \mathcal{S} . The estimated sound source direction is then $\tilde{\mathbf{x}} = \mathbf{x}_{\tilde{n}}$ where $\tilde{n} = \underset{n \in [1:N]}{\operatorname{argmin}} C_{\mathcal{S}}(n)$.

The accuracy of such methods will directly depend on the number of positions learned since they do not interpolate between learned positions. Moreover, since they make use of the entire training dataset to localize sounds, they will require an $\mathcal{O}(DN)$ memory storage. We will consider cost functions that are continuous with respect to interaural feature values. This means that interaural feature vectors in a neighborhood of a learned vector \mathbf{y}_n will be mapped to the same position \mathbf{x}_n , and hence treated similarly as \mathbf{y}_n . For this reason, we refer to such methods as *piecewise-constant mapping*. Indeed the function eventually used to map interaural features to positions is piecewise-constant, which amounts to make a piecewise-constant approximation of the true binaural manifold. Three cost functions are considered in the remainder of this section.

4.2.1 Unweighted cost function

A straightforward cost function allowing to deal with spectrogram inputs is:

$$C_{\mathcal{S}}^1(n) = \sum_{d=1}^D \sum_{t=1}^T \chi_{dt} (y'_{dt} - y_{dn})^2. \quad (4.1)$$

It corresponds to summing the squared differences between observed spectrogram values and their corresponding training feature value. However, interaural feature vectors may contain IPD values, which are angles in the $] - \pi, \pi]$ circle. For such values, the use of squared differences is not appropriate. For example, if ϵ is a small positive value, the two angles $(-\pi + \epsilon/2)$ and $(\pi - \epsilon/2)$ will have a large squared difference, although their actual distance on the circle is small. We will thus treat differently real (ILD) values and angular (IPD) values. For convenience, we introduce the binary function Δ defined by $\Delta(y_1, y_2) = y_1 - y_2$ when y_1 and y_2 represent real values and by $\Delta(y_1, y_2) = \arg(e^{j(y_1 - y_2)}) \in] - \pi, \pi]$ when y_1 and y_2 represent angular values. We accordingly redefine the cost function $C_{\mathcal{S}}^1$ as

$$C_{\mathcal{S}}^1(n) = \sum_{d=1}^D \sum_{t=1}^T \chi_{dt} \Delta(y'_{dt} - y_{dn})^2 \quad (4.2)$$

4.2.2 Normalized cost function

C_S^1 gives equal weights to all binaural features at all frequency channels. This might not be the best choice in practice, since features may have different *scales*. For instance, IPD values range in $]-\pi, \pi]$ while ILD values in dB typically range in $[-30, 30]$. To avoid this issue, each feature can be normalized by its variance in the training set. This yields the following cost function:

$$C_S^2(n) = \sum_{d=1}^D \frac{N}{\sum_{m=1}^N \Delta(y_{dm} - \bar{y}_d)^2} \sum_{t=1}^T \chi_{dt} \Delta(y'_{dt} - y_{dn})^2 \quad (4.3)$$

where \bar{y}_d denotes the mean value of $\{y_{dn}\}_{n=1}^N$. Again, a distinction must be made to calculate this mean, depending on whether d is the index of a real value or an angular value. Means of real values are calculated with the standard mean $\bar{y}_d = 1/N \sum_{n=1}^N y_{dn}$ while means of angular values are calculated with the *angular mean* $\bar{y}_d = \arg(1/N \sum_{n=1}^N e^{jy_{dn}})$.

4.2.3 Probabilistic Piecewise Constant Mapping

Neither C_S^1 nor C_S^2 takes into account the different amounts of noise in different features and frequency channels. For example, ILD features at low frequencies are more noisy due to background noise, and hence less reliable for sound source localization than other features. To account for this, we propose a probabilistic model. We denote by $\mathbf{g} : \mathcal{X} \rightarrow \mathcal{Y} \subset \mathbb{R}^D$ the mapping function from source position to interaural feature, such that $\mathbf{y}_n = \mathbf{g}(\mathbf{x}_n)$ for all $(\mathbf{x}_n, \mathbf{y}_n) \in \mathcal{T}$. We use the decomposition $\mathbf{g} = (g_1, \dots, g_D)^\top$ such that $y_{dn} = g_d(\mathbf{x}_n)$ for all $d \in [1 : D]$ and $n \in [1 : N]$. Let us assume that every non-missing interaural spectrogram value y'_{dt} is the image by g_d of an unknown sound source position \mathbf{x} , perturbed by Gaussian noise with variance ρ_d^2 . The probability density of y'_{dt} given \mathbf{x} and ρ_d^2 writes:

$$p(y'_{dt}; \boldsymbol{\psi}) = \mathcal{L}\mathcal{N}(y'_{dt}; g_d(\mathbf{x}), \rho_d^2) \quad (4.4)$$

where $\boldsymbol{\psi} = \{\{\rho_d^2\}_{d=1}^D, \mathbf{x}\}$ denotes the model's parameters and $\mathcal{L}\mathcal{N}$ denotes the standard normal distribution adapted to possible angular values, *i.e.*,

$$\mathcal{L}\mathcal{N}(x; \mu, \sigma^2) = \mathcal{N}(\Delta(x, \mu); 0, \sigma^2). \quad (4.5)$$

As mentioned in [Mandel 10], (4.5) approximates well the normal distribution on the circle $]-\pi, \pi]$ when σ is small relative to 2π , which is generally the case in practice. For larger values of σ , the distribution becomes close to uniform on $]-\pi, \pi]$.

We assume that all the spectrogram observations are independent and identically distributed (iid). Note that this does not contradict the well-known dependency between interaural features at different frequency channels induced by the HRTFs. Indeed, only the *noises* term perturbing these different features are supposed independent. The log-likelihood of observed data $\mathcal{S} = \{\{\mathbf{y}'_{dt}\}_{d=1, t=1}^{D, T}, \mathcal{X}\}$ using model (4.4) writes:

$$\mathcal{L}_{\text{PPCM}}(\mathcal{S}; \boldsymbol{\psi}) = \sum_{\chi_{dt}=1} \log p(y'_{dt} | \mathbf{x}, \rho_d^2)^{\chi_{dt}} \quad (4.6)$$

where $\sum_{\chi_{dt}=1} \{.\}$ denotes the sum over all d and t verifying $\chi_{dt} = 1$. We want to find parameters $\tilde{\psi} = \{\tilde{\mathbf{x}}, \{\tilde{\rho}_d^2\}_{d=1}^D\}$ maximizing $\mathcal{L}(\mathcal{S}; \psi)$. By finding zeros of the derivative, noise variances $\tilde{\rho}_d^2$ can be expressed in closed form as a function of $\tilde{\mathbf{x}}$:

$$\tilde{\rho}_d^2 = \frac{1}{\bar{\chi}_d} \sum_{t=1}^T \chi_{dt} \Delta(y'_{dt} - g_d(\tilde{\mathbf{x}}))^2 \quad \text{where} \quad \bar{\chi}_d = \sum_{t=1}^T \chi_{dt}. \quad (4.7)$$

By substituting this expression in the log-likelihood, it follows that $\tilde{\mathbf{x}}$ is a minimum of the following expression:

$$\sum_{d=1} \bar{\chi}_d \log \left(1 + \frac{\Delta(\bar{y}'_d, g_d(\mathbf{x}))^2}{1/\bar{\chi}_d \sum_{t=1}^T \chi_{dt} \Delta(y'_{dt}, \bar{y}'_d)^2} \right) \quad \text{where} \quad \bar{y}'_d = \frac{1}{\bar{\chi}_d} \sum_{t=1}^T \chi_{dt} y'_{dt}. \quad (4.8)$$

Since \mathbf{g} is only known on the discrete support $\{\mathbf{x}_n\}_{n=1}^N$, maximizing this expression with respect to $\mathbf{x} \in \mathcal{X}$ is not directly possible. However, following the idea of a piecewise-constant approximation of \mathbf{g} , we can look for an optimal value in $\{\mathbf{x}_n\}_{n=1}^N$. Since $y_{dn} = g_d(\mathbf{x})$ for all $d \in [1 : D]$ and $n \in [1 : N]$, this amounts to finding $\tilde{n} \in [1 : N]$ minimizing the following discrete cost function:

$$C_S^3(n) = \sum_{d=1} \bar{\chi}_d \log \left(1 + \frac{\Delta(\bar{y}'_d, y_{dn})^2}{1/\bar{\chi}_d \sum_{t=1}^T \chi_{dt} \Delta(y'_{dt}, \bar{y}'_d)^2} \right) \quad (4.9)$$

and set $\tilde{\mathbf{x}} = \mathbf{x}_{\tilde{n}}$.

The cost function C_S^3 present several important advantages compared to C_S^1 and C_S^2 . First, it relies on a probabilistic model. This will allow to extend the method to the case of mixture of sound sources in section 5.3 of chapter 5. Second, it weights the squared distance between training and observed features according to their variance along time rather than their scale. It means that the method will put more “trust” in features that are relatively close to their mean along time, or in other words, that are less noisy. This choice seems more relevant, and showed to yield much better sound source localization performance on preliminary tests. Finally, C_S^3 offers an interesting computational property. Contrary to C_S^1 and C_S^2 , it only depends on the temporal means $\{\bar{y}'_d\}_{d=1}^D$ of observed binaural features, which can be computed beforehand. Therefore, the computational complexity to minimize C_S^3 is $\mathcal{O}(DN + TN)$, while the computational complexity to minimize C_S^1 and C_S^2 is $\mathcal{O}(DTN)$. For large training set, *e.g.* $N = 10,800$ points in the audio-motor training set presented in section 2.3.2, the memory and time costs of C_S^1 and C_S^2 become prohibitive, making them unusable. In contrast, C_S^3 is several orders of magnitude faster and more accurate.

For these reasons, the cost function C_S^3 and associated probability model will be used for piecewise constant mapping. Will refer to the associated technique as *probabilistic piecewise constant mapping* (PPCM).

4.3 Probabilistic Piecewise Affine Mapping for Spectrograms

In section 3.6.5, we already demonstrated the ability of Gaussian locally-linear mapping (GLLiM) models to accurately map interaural feature vectors obtained from white noise recordings to source positions. However, the GLLiM inverse mapping functions given in (3.12) only allows to map an input vector to an output vector. As detailed in section 4.1 spectrograms of natural sound sources are noisy, redundant time series of vectors with missing values. A great advantage of the Bayesian framework used in GLLiM is that these models can be easily extended to deal with such situations. In this section, we extend the GLLiM inverse mapping formula (3.12) to spectrogram inputs in the case of diagonal and equal noise covariance matrices, *i.e.*,

$$\Sigma_k = \text{diag}(\sigma_1^2, \dots, \sigma_d^2). \quad (4.10)$$

This amounts to assume that the noise perturbing the different ILD and IPD features at different frequency channels are independent and with equal variance in all affine components for a given feature d . As in previous section, this does not contradict the dependency between interaural features since only the noises are supposed independent. The general resulting sound source localization method will be referred to as *spectrogram inversion*. The derivations of this section are valid for any GLLiM model verifying (4.10), including PLOM models with a strictly positive latent variable dimension L_w (see section 3.4). However, the PPAM (PLOM-0) model will be used in practice as adding latent components did not significantly improve sound source localization using our datasets (see preliminary experiments in section 3.6.5). The spectrogram inversion method for PPAM will be referred to as *inverse PPAM* (iPPAM).

Let $\tilde{\theta}$ denote a set of PPAM parameters trained from a training set $\mathcal{T} = \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^N$. Recall that PPAM cannot deal with circular data (see section 3.6.5). Hence IPD values need to be expressed in \mathbb{C} or equivalently \mathbb{R}^2 . Therefore, contrary to section 4.2, no distinction will be made between IPD or ILD feature values in \mathbf{y} . Let $\mathcal{S} = \{\{\mathbf{y}'_1, \dots, \mathbf{y}'_T\}, \chi\}$ be an observed interaural spectrogram. The following key theorem holds:

Theorem 2 *Under constraint (4.10), if we suppose that all the observations in spectrogram \mathcal{S} are assigned to the same sound source position and the same local affine transformation, the posterior distribution $p(\mathbf{x}|\mathcal{S}; \tilde{\theta})$ is a Gaussian mixture model in \mathbb{R}^L , *i.e.*,*

$$p(\mathbf{x}|\mathcal{S}; \tilde{\theta}) = \sum_{k=1}^K \nu_k \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_k, \mathbf{V}_k). \quad (4.11)$$

Parameters $\{\boldsymbol{\mu}, \mathbf{V}, \nu_k\}$ can be expressed in closed-form with respect to learned parame-

ters $\tilde{\theta}$ and input data \mathcal{S} :

$$\boldsymbol{\mu}_k = \mathbf{V}_k \left(\tilde{\boldsymbol{\Gamma}}_k^{-1} \tilde{\mathbf{c}}_k + \sum_{d,t=1}^{D,T} \frac{\chi_{dt}}{\tilde{\sigma}_d^2} \tilde{\mathbf{a}}_{dk} (y'_{dt} - \tilde{b}_{dk}) \right), \quad (4.12)$$

$$\mathbf{V}_k = \left(\tilde{\boldsymbol{\Gamma}}_k^{-1} + \sum_{d,t=1}^{D,T} \frac{\chi_{dt}}{\tilde{\sigma}_d^2} \tilde{\mathbf{a}}_{dk} \tilde{\mathbf{a}}_{dk}^\top \right)^{-1} \quad \text{and} \quad (4.13)$$

$$\nu_k \propto \tilde{\pi}_k \frac{|\mathbf{V}_k|^{\frac{1}{2}}}{|\tilde{\boldsymbol{\Gamma}}_k|^{\frac{1}{2}}} \exp \left(-\frac{1}{2} \left(\sum_{d,t=1}^{D,T} \frac{\chi_{dt}}{\tilde{\sigma}_d^2} (y'_{dt} - \tilde{b}_{dk})^2 + \tilde{\mathbf{c}}_k^\top \tilde{\boldsymbol{\Gamma}}_k^{-1} \tilde{\mathbf{c}}_k - \boldsymbol{\mu}_k^\top \mathbf{V}_k^{-1} \boldsymbol{\mu}_k \right) \right) \quad (4.14)$$

where ν_k is normalized to sum to 1 over k and we used the decompositions:

$$\tilde{\mathbf{A}}_k = (\tilde{\mathbf{a}}_{1k}, \dots, \tilde{\mathbf{a}}_{Dk})^\top \quad \text{with } \mathbf{a}_{dk} \in \mathbb{R}^L \quad \text{and} \quad (4.15)$$

$$\tilde{\mathbf{b}}_k = (\tilde{b}_{1k}, \dots, \tilde{b}_{Dk})^\top \quad \text{with } b_{dk} \in \mathbb{R}. \quad (4.16)$$

A proof of Theorem 2 is provided in Appendix 4.A. It generalizes the GLLiM inverse conditional density (3.9), which corresponds to the unique, complete observation case, *i.e.*, $T = 1, \chi = \mathbf{1}$. As in (3.11), the posterior expectation can be used to obtain an estimate $\hat{\mathbf{x}}$ of the sound source position given a spectrogram input \mathcal{S} and learned parameters $\tilde{\theta}$:

$$\hat{\mathbf{x}} = \mathbb{E}[\mathbf{x} | \mathcal{S}; \tilde{\theta}] = \sum_{k=1}^K \nu_k \boldsymbol{\mu}_k. \quad (4.17)$$

Alternatively, one could use the full posterior distribution and, for instance, combine it with other external probabilistic knowledge to increase the localization accuracy or extract higher order information.

The computational cost of localizing a sound source with this method is $\mathcal{O}(DTK)$. The memory cost is the size of learned parameters $\tilde{\theta}$, *i.e.*, $\mathcal{O}(DK)$. Note that while the time and memory costs of PPCM presented in section 4.2 are proportional to the training set size N , the costs of iPPAM *do not depend on* N and are proportional to the number of affine transformations K . In other words, iPPAM reduces the training data size from N to K , which may be advantageous when dealing with very large training set.

4.4 Sound Source Localization Results

4.4.1 Audio-Motor training set

We first evaluate the proposed sound source localization algorithms PPCM and iPPAM using the audio-motor dataset presented in section 2.3.2. For simplicity, we identify azimuth-elevation source directions to pan-tilt motor states in this section¹. Recall that

¹Localization errors can be measured independently in both spaces, since only a slight distortion exist between them, *i.e.*, (2.5).

the audio-motor dataset contain 10,800 sound source positions spanning the azimuth-elevation space $\mathcal{X}_C = [-180^\circ, 180^\circ] \times [-60^\circ, 60^\circ]$ which has a cylinder topology. However, PPAM requires to estimate a Gaussian mixture over the sound source position space. This can only be done if this space has a Euclidean (planar) topology. For this reason, only the 9,600 source positions in $\mathcal{X}_P = [-160^\circ, 160^\circ] \times [-60^\circ, 60^\circ]$ will be used in practice, which corresponds to a planar space. One way to extend PPAM to a non-planar low-dimensional space would be to cut the space into 2 or more planar parts. A PPAM model could then be learned separately on each planar part, and we could ultimately consider a mixture of these PPAM models. This extension however, is left for future work.

In a first experiment, a subset of 9,100 position-to-interaural-feature associations obtained from white noise recordings was used as training set for the two algorithms. They were then tested on speech sources emitting from the remaining 500 source positions, so that tested positions were outside of the training set. The test speech recordings were all cut to last 1 second. The average amount of missing data in test spectrograms was $\approx 80\%$. iPPAM's only parameter K was manually set to 300. Different types of feature were used in training sets: ILD only, IPD only or a concatenation of both (ILPD). For each type of feature, we considered two situations: (i) Training interaural feature vectors are *clean*, *i.e.* we take the *temporal mean* of each white-noise interaural spectrogram and (ii) training interaural feature vectors are *noisy*, *i.e.*, we extract *one vector* from each white-noise interaural spectrogram. As explained in section 2.2.2, taking the temporal mean allows to drastically reduce noise in interaural feature vectors. However, it requires to have a longer white-noise recording from each position (1 second in this case). On the other hand, individual vectors corresponding to a single spectrogram window are faster to obtain, but a lot noisier.

For comparison, we used the baseline sound source localization method PHAT histogram [Aarabi 02] on the same test sounds. PHAT estimates the sound source's *time difference of arrival* (TDOA), by accumulating cross-correlations at different time and frequency channels. Note that this method estimates the sound source's *time difference of arrival* (TDOA), which can only be mapped to a frontal azimuth (1D) angle. Indeed, TDOAs induce front-back localization ambiguities. Therefore only sources emitting from $[-90^\circ, 90^\circ] \times [-60^\circ, 60^\circ]$ were used to evaluate PHAT. A linear regressor was trained to map TDOA values to azimuth angles using the white noise training data². The few existing 2D sound source localization methods in the literature, *e.g.*, [Kullaib 09], could not be used for comparison. Indeed, [Kullaib 09] relies on artificial ears with a spiral shape.

Sound source localization errors obtained with PHAT, PPCM and iPPAM using all training vectors considered, namely ILPD, ILD, IPD (clean) and ILPD, ILD, IPD (noisy) are showed in table 4.1. PPCM and iPPAM dramatically outperform PHAT in azimuth localization, with no outliers as opposed to 30% of outliers using PHAT. They also provide a very high localization accuracy in elevation. The poor results obtained with PHAT may be explained by important variations in elevation in that training set, while TDOA can only be accurately mapped to azimuth if the elevation is near 0° . Unsurprisingly, PPCM and iPPAM generally perform best when using combined ILD and IPD features.

²A linear dependency was observed in practice.

Table 4.1: Localization error average and standard deviation in degrees (Avg±Std), percentage of outliers (Out) and mean localization time in seconds of a 1 second speech source using PPCM and PPAM ($K = 300$) on different training vectors and the baseline method PHAT [Aarabi 02]. Avgs and Stds are calculated over inlying estimates only, among 500 speech recordings randomly picked from the audio-motor test set. Estimates are considered outliers if their distance to ground truth in the azimuth-elevation space is higher than 45° .

Method	PPCM				PPAM			
	Training vectors	Azimuth	Elevation	Out	Time	Azimuth	Elevation	Out
ILPD (clean)	1.16±1.0	0.95±1.0	0.0	.63	1.42±1.2	1.25±1.1	0.0	3.6
ILD (clean)	2.83±2.9	1.71±1.8	1.4	.31	2.82±2.2	1.74±1.5	0.0	1.2
IPD (clean)	1.01±1.0	1.21±1.0	0.0	.32	1.54±1.6	1.74±1.5	0.0	2.4
ILPD (noisy)	1.57±1.2	1.26±1.2	0.0	.63	1.48±1.3	1.25±1.1	0.0	3.6
ILD (noisy)	2.84±2.8	1.72±1.8	1.2	.32	2.39±2.0	1.50±1.2	0.0	1.2
IPD (noisy)	1.46±1.2	1.42±1.2	0.0	.32	1.00±1.0	1.11±1.0	5.6	1.4

Baseline	Azimuth	Out	Time
PHAT [Aarabi 02]	5.00±6.3	30	0.4

The two algorithms performed similarly on this dataset. It is worth noting that while PPCM slightly outperforms iPPAM using the clean training sets, the contrary is observed using the noisy training set. Moreover, while the largest localization error obtained with iPPAM was 22° (no outlier), PPCM made a few very large errors (more than 100°) using ILD only. This suggests that PPCM is less robust than PPAM, and perhaps less suited for real-world applications. Note that PPAM approximates binaural manifolds with only 300 affine components, while PPCM require 9,000 points. The quantity of information required by PPCM makes it more prone to overfitting. Moreover, PPCM always estimated the position with at least 2° error in either azimuth or elevation, because test positions were outside the training set. This issue is not observed with PPAM, since it interpolates between training points.

Table 4.1 also shows the average time needed by the different methods to localize a 1 second sound. All methods were implemented in Matlab and run on a standard laptop. They all achieve the localization of a 1 second sound in the order of 1 second using these basic implementations, showing that they are all suitable for real-time applications.

We then further studied the influence of K and N on the sound source localization results obtained with PPCM and iPPAM, using the audio-motor training set with clean feature vectors. As showed in Figure 4.2(a), the only parameter K of PPAM can be tuned based on a trade-off between computational time and accuracy: high-values of K require more time but yield more accurate results. Choosing $K = 50$ instead of $K = 300$ divided by 6 the computation time, while increasing the mean localization error by 2.4° only, without adding outliers (maximum error of 23.4°). The computation time to localize a 1 second source is then below 0.6 second, making the method suitable for real time. Although there are omitted for clarity in Figure 4.2(a), we obtained very high localization errors (59° distance in average) using $K = 1$. This confirms the results of section 2.4, showing that interaural data lie on a locally-linear rather than linear manifold. Figure 4.2(b), confirms that iPPAM's computational time is not affected by the training set size,

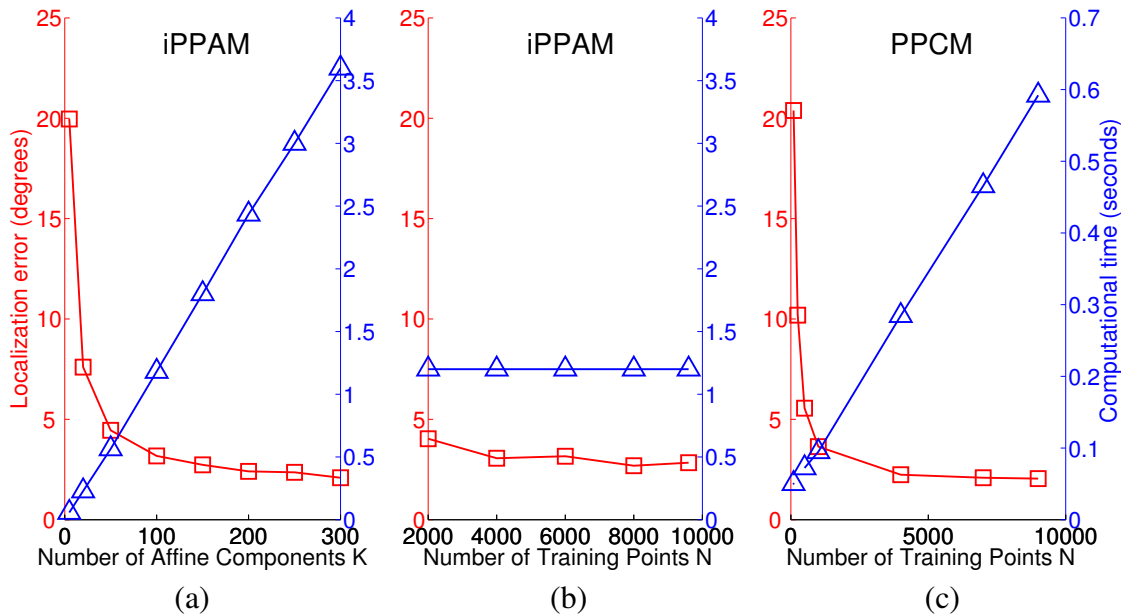


Figure 4.2: Influence of K and N on iPPAM's and PPCM's mean localization time and azimuth-elevation distance error, using the audio-motor training set. Both algorithms are trained using clean ILPD features. (a) PPAM is trained on the complete set ($N = 9600$) and K varies from 5 to 300. Means are computed over 500 tests at each point. (b) PPAM is trained on a random subset of N points, N from 2000 to 9600, and K is fixed to 100. Means are computed over 200 localization tasks and 4 different training at each point. (c) PPCM is used with random subsets of the complete training set, with N from 100 to 9100. A different random subset is used for each localization task, and test sounds are emitted from out-of-training positions. Means are computer over 500 tests at each point.

since only learned parameters are kept for localization. The figure also reveals that the specific training set used does not affect much iPPAM's results for a fixed K ³. Using 2,000 instead of 9,600 training points with $K = 100$ increased the error by 1.2° only, without creating outliers. However, a training set of 2,000 points is 5 times faster to obtain using the audio-motor sampling technique described in section 2.3.2.

The contrary is observed for PPCM in Figure 4.2(c). Since PPCM relies on the entire training set to localize sounds, both its accuracy and computational time are strongly influenced by the training set size N . The linear increase of computational time in N suggests that PPCM may not be scalable to very large training sets. The large decrease of performance for smaller values of N shows that PPCM require dense training sets. In conclusion from these figures, while PPCM slightly outperforms iPPAM in terms of localization accuracy when using a dense and clean training set, it appears to be less robust to the specific training set used than PPAM. On the other hand, iPPAM's performances do not seem to be much influenced by the training set. This may make PPAM more suitable than PPCM for more realistic data, as addressed in the next section.

³However, as explained in section 3.6, when the number of training points per affine component becomes too small (typically less than 20 points per components) some components become empty and are removed along the iterations of the EM algorithm, thus reducing K .

Table 4.2: Audio-visual localization results. Localization error average and standard deviation in pixels (Avg \pm Std), percentage of outliers (Out) and average localization time of a 1 second sound in seconds using PPCM and PPAM ($K = 32$) with different features and the baseline method PHAT [Aarabi 02]. Avgs and Stds are calculated over inlying estimates only, among 107 speech recordings from the audio-motor test set. Estimates are considered outliers if their horizontal or vertical error is higher than 150 pixels.

Method	PPCM				PPAM			
Features used	Horizontal	Vertical	Out	Time	Horizontal	Vertical	Out	Time
ILPD	18.2 \pm 16	15.0 \pm 18	7.5	.07	21.9 \pm 17	23.1 \pm 20	0.0	.23
ILD	20.2 \pm 19	19.4 \pm 25	19	.03	26.1 \pm 19	21.1 \pm 19	1.9	.08
IPD	17.4 \pm 15	16.8 \pm 22	6.5	.03	24.0 \pm 21	28.9 \pm 28	2.8	.15

Baseline	Azimuth	Out	Time
PHAT	43.8 \pm 29	19	0.4

4.4.2 Audio-Visual training set

Results obtained with the audio-motor dataset validate the ability of both PPCM and iPPAM to localize sound sources emitting from a very wide range of positions, including low and high elevations and sources emitting from behind the listener. It also show their potential to reach an unequaled 2D localization precision when the dataset is sufficiently large. However, one may view these results as a proof of concept only. Indeed, as mentioned in section 2.3.2, a recording made at a given motor-state only approximates what would be perceived if the source was actually moved to the corresponding relative position in the room. This approximation holds only if the room presents relatively few asymmetries and reverberations, which may or may not be the case in practice. To confirm that both PPCM and iPPAM perform accurate sound source localization in real world conditions, we now test them using the smaller but more realistic audio-visual training set presented in section 2.3.3. PPCM and iPPAM were trained with mean interaural feature vectors obtained from white noise recordings. We used the complete audio-visual training set containing $18 \times 24 = 432$ positions in the 480×640 camera image. The algorithms were trained using either ILD features only, IPD only or both (ILPD). They were then tested on 107 speech recordings⁴, covering an 9×12 regular grid in the image. Recall that 30 pixels roughly correspond to 1.3° . Again, the test speech recordings were all cut to last 1 second and the average amount of missing data in test spectrograms was $\approx 80\%$.

Table 4.2 shows horizontal and vertical localization errors in pixels obtained using PPCM, iPPAM and PHAT. A linear regressor was trained to map TDOA values onto horizontal pixel coordinates using white noise training data⁵. iPPAM was trained with $K = 32$ in this experiment. Again, both PPCM and PPAM significantly outperform PHAT. Although mean inlying PPCM errors are slightly smaller than iPPAM errors, the maximum horizontal or vertical error returned by PPAM using ILPD features was 90 pixels, while 8 out of 108 estimates had more than 150 pixels horizontal or vertical error using PPCM. This observation is confirmed by figures 4.3(a) and 4.3(b). While varying

⁴One faulty test recording out of 108 was removed due to undesired movements of the emitter.

⁵A linear dependency was observed in practice.

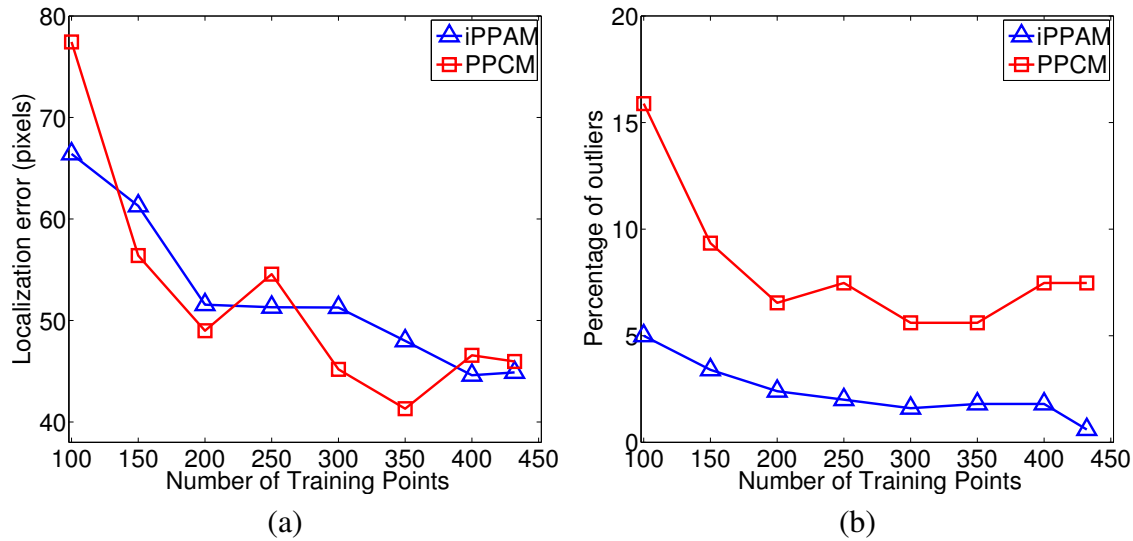


Figure 4.3: Influence of N (from 100 to 432) on iPPAM ($K = 10$) and PPCM's performance using the audio-visual training set. (a) Mean pixel distance error. (b) Percentage of outliers. Outliers are defined by estimates with more than 150 pixel horizontal or vertical error. For each value of N both algorithms were tested on 107 speech signals. iPPAM was used with 5 different set of parameters learned from different random subsets of the full training set. PPCM was used with a different random subset of the full training set for each test.

the training set size from 100 to 432 points, PPCM and PPAM's mean errors are similar, while PPCM yielded 3 to 12 times more outliers than PPAM. Note that in this experiment, PPAM with $K = 10$ performed better than PPCM on the full training set, while the size of PPAM's learned parameters $\tilde{\theta}$ was 14 times smaller than the full training set's size.

Experiments on audio-visual data confirm the remarks of previous section, *i.e.*, PPCM and iPPAM show similar performances in terms of mean error, but PPCM returns more outliers using noisy and small training set. In the audio-visual training set, the manual positioning of the emitter around the listener may involve slight changes in orientation and distance between test and training data. These can be handled by the interpolating ability of iPPAM, but cannot be handled by PPCM. The former is thus more adapted to realistic audio-visual scenarios.

4.4.3 Localization of a human speaker in realistic conditions

We finally tested iPPAM on a realistic auditory scenes analysis task. After gathering the audio-visual training set, an audio-visual scenario involving a human speaker was recorded with the same setup placed in the same position in the same room. A participant is asked to come to the field of view of the camera and counts to 20. The speaker is static while pronouncing each number, and places his head at a different position in the image for each pronounced number. The binaural sound track and video are synchronized using hand claps at the beginning and at the end of the recording. Note that this is a particularly challenging scenario for several reasons. First, the speaker places himself

at different distances than the training distance. Second his head has different orientations than the loud-speaker during training, and more generally, human speakers do not emit sounds with the same directionality as loud-speakers. This may change the recorded reverberations for a given position. Third, the participant may perform some slight head translations and/or rotations while speaking, which may perturb recorded interaural features. Finally, the speaker emits shorter and less loud sounds than in the training set, which reduces the number of available data.

We trained PPAM with the audio-visual white-noise ILPD training set using $K = 32$. Then, iPPAM was run on a 720ms sliding analysis window over the counting scenario soundtrack, in order to estimate a position at each video frame. iPPAM is run only when



Figure 4.4: Localization results with iPPAM ($K = 32$) using the audio-visual white-noise dataset for training. The speaker counts from 1 to 20 (white numbers) with a normal voice loudness and is static while pronouncing each number. The center of the red circle is the estimated source position by iPPAM on a 720ms audio analysis window centered on the displayed video frame.

enough time-frequency points are observed in the analysis window, *i.e.*, more than 3,000 non-missing point in the spectrogram. The source position estimated at each video frame can then be plotted in the corresponding camera image. For each number pronounced by the speaker, Figure 4.4 shows an image extracted from the video marked with the estimated source position. Images selected are chosen around the middle of the utterance, because the first and last frames were sometimes wrong due to lack of enough data in the sliding analysis window. As can be seen, the algorithm localizes a sound source near the mouth of the speaker for all pronounced number. The largest mouth-to-estimate distance is 128 pixels, made with pronounced number 3. It corresponds to $\approx 1.7^\circ$ error in azimuth and $\approx 5.3^\circ$ error in elevation. Note that considering computational times reported in table 4.2, this on-line sound source localization method is ready for real-time implementation.

4.5 Conclusion

In this chapter, we addressed the long-studied problem of binaural sound source localization through *supervised learning*. This approach strongly contrasts with traditional approaches in sound source localization which usually assume the mapping known, based on simplified sound propagation models, *e.g.* [Aarabi 02, Yılmaz 04, Kullaib 09, Liu 10, Alameda-Pineda 12]. We proposed two methods, PPCM and iPPAM, which provided very high sound source localization accuracy, with mean azimuth angular errors more than 2 times lower than the baseline PHAT histogram, and with much less outliers. In addition, unlike existing TDOA-based sound source localization methods, PPCM and iPPAM accurately estimate the elevation angle of the sound source. They are also able to localize sound source coming from a wide range of directions, including high and low elevation or sources coming from the back (no front-back ambiguity). Such a high 2D localization accuracy from binaural recordings in real world conditions is unequalled to date, to the best of our knowledge. Accurate localization could be useful in many applications, *e.g.*, speaker identification in a crowded audio-visual scene for robotic [Cech 13b], or hearing aid. Moreover, the probabilistic formulation of both PPCM and iPPAM will allow for EM-based extensions to multiple sound sources localization and separation in the next chapter.

Appendix

4.A Proof of Theorem 2

In this Appendix, we prove Theorem 2 by detailing the derivation of

$$p(\mathbf{x}|\mathcal{S};\tilde{\boldsymbol{\theta}}) \quad (4.18)$$

i.e., a probabilistic mapping from spectrogram input $\mathcal{S} = \{y'_{dt}, \chi_{dt}\}_{d=1,t=1}^{D,T}$ to source position \mathbf{x} based on learned GLLiM parameters $\tilde{\boldsymbol{\theta}}$. If we include the hidden assignment variable Z in (4.18) using the integration rule, we obtain:

$$p(\mathbf{x}|\mathcal{S};\tilde{\boldsymbol{\theta}}) = \sum_{k=1}^K p(\mathbf{x}|\mathcal{S}, Z = k; \tilde{\boldsymbol{\theta}})p(Z = k|\mathcal{S}; \tilde{\boldsymbol{\theta}}). \quad (4.19)$$

Since, by definition, GLLiM models imply an affine dependency between variables \mathbf{X} and \mathbf{Y} for a given Z , and both \mathbf{X} and \mathbf{Y} are Gaussian given Z , the term $p(\mathbf{x}|\mathcal{S}, Z = k; \tilde{\boldsymbol{\theta}})$ is a Gaussian distribution in \mathbf{x} . In other words, for each k , there is a mean $\boldsymbol{\mu}_k \in \mathbb{R}^L$ and a covariance matrix $\mathbf{V}_k \in \mathbb{R}^{L \times L}$ such that $p(\mathbf{x}|\mathcal{S}, Z = k; \tilde{\boldsymbol{\theta}}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_k, \mathbf{V}_k)$.

The term $p(Z = k|\mathcal{S}; \tilde{\boldsymbol{\theta}})$ do not depend on \mathbf{x} and will be denoted ν_k . With these notations, (4.19) leads directly to the desired result (4.11). We now detail the calculation of the GMM's parameters $\{\boldsymbol{\mu}_k, \mathbf{V}_k, \nu_k\}_{k=1}^K$.

Calculation of $\boldsymbol{\mu}_k$ and \mathbf{V}_k Using Baye's inversion formula we have

$$p(\mathbf{x}|\mathcal{S}, Z = k; \tilde{\boldsymbol{\theta}}) = \frac{p(\mathcal{S}|\mathbf{x}, Z = k; \tilde{\boldsymbol{\theta}})p(\mathbf{x}|Z = k; \tilde{\boldsymbol{\theta}})}{p(\mathcal{S}|Z = k; \tilde{\boldsymbol{\theta}})}. \quad (4.20)$$

The assumption (4.10) means that noises on the different observations in \mathcal{S} are independent and identically distributed. Therefore, by omitting the denominator of (4.20) which does not depend on \mathbf{x} , we can write:

$$p(\mathbf{x}|\mathcal{S}, Z = k; \tilde{\boldsymbol{\theta}}) \propto \left\{ \prod_{d=1,t=1}^{D,T} p(y'_{dt}|\mathbf{x}, Z = k; \tilde{\boldsymbol{\theta}})^{\chi_{dt}} \right\} p(\mathbf{x}|Z = k; \tilde{\boldsymbol{\theta}}) \quad (4.21)$$

$$= \left\{ \prod_{d=1,t=1}^{D,T} \mathcal{N}(y'_{dt}|\tilde{\mathbf{a}}_{dk}^\top \mathbf{x} + \tilde{b}_{dk}, \tilde{\sigma}_d^2)^{\chi_{dt}} \right\} \mathcal{N}(\mathbf{x}; \tilde{\mathbf{c}}_k, \tilde{\boldsymbol{\Gamma}}_k) \quad (4.22)$$

$$= \frac{C}{|\tilde{\boldsymbol{\Gamma}}_k|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} \left(\sum_{d=1,t=1}^{D,T} \frac{\chi_{dt}}{\tilde{\sigma}_d^2} (y'_{dt} - \tilde{\mathbf{a}}_{dk}^\top \mathbf{x} - \tilde{b}_{dk})^2 + (\mathbf{x} - \tilde{\mathbf{c}}_k)^\top \tilde{\boldsymbol{\Gamma}}_k^{-1} (\mathbf{x} - \tilde{\mathbf{c}}_k) \right) \right\} \quad (4.23)$$

where C does not depend on \mathbf{x} or k . Since we know that $p(\mathbf{x}|\mathcal{S}, Z = k; \tilde{\boldsymbol{\theta}})$ is a normal distribution in \mathbf{x} with mean $\boldsymbol{\mu}_k$ and covariance \mathbf{V}_k , we can identify the term within the exponential (4.23) to

$$-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^\top \mathbf{V}_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k). \quad (4.24)$$

By one-to-one identification of the constant, linear, and quadratic terms in \mathbf{x} in the exponential (4.23) and in (4.24), we obtain the desired formulas (4.12) and (4.13) for $\boldsymbol{\mu}_k$ and \mathbf{V}_k ■.

Calculation of ν_k Using Baye's inversion formula we obtain:

$$\nu_k = p(Z = k|\mathcal{S}; \tilde{\boldsymbol{\theta}}) = \frac{p(\mathcal{S}|Z = k; \tilde{\boldsymbol{\theta}})p(Z = k; \tilde{\boldsymbol{\theta}})}{\sum_{j=1}^K p(\mathcal{S}|Z = j; \tilde{\boldsymbol{\theta}})p(Z = j; \tilde{\boldsymbol{\theta}})} \quad (4.25)$$

$$\propto \tilde{\pi}_k p(\mathcal{S}|Z = k; \tilde{\boldsymbol{\theta}}) \quad (4.26)$$

Unfortunately, we cannot directly decompose $p(\mathcal{S}|Z = k; \tilde{\boldsymbol{\theta}})$ into a product over (d, t) , as done with $p(\mathcal{S}|\mathbf{x}, Z = k; \tilde{\boldsymbol{\theta}})$ in previous paragraph. Indeed, while (4.10) means that the different observations in the spectrograms are independent *given \mathbf{x} and Z* , this is not true for the same observations *given Z only*. However, we can use (4.20) to obtain

$$p(\mathcal{S}|Z = k; \tilde{\boldsymbol{\theta}}) = \frac{p(\mathcal{S}|\mathbf{x}, Z = k; \tilde{\boldsymbol{\theta}})p(\mathbf{x}|Z = k; \tilde{\boldsymbol{\theta}})}{p(\mathbf{x}|\mathcal{S}, Z = k; \tilde{\boldsymbol{\theta}})}. \quad (4.27)$$

The numerator is given by (4.23) and the denominator is the normal $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_k, \mathbf{V}_k)$. After simplification of the terms in \mathbf{x} we obtain the desired formula (4.14) for ν_k ■.

CHAPTER 5

MULTIPLE SOUND SOURCES SEPARATION AND LOCALIZATION

This chapter proposes three novel methods for supervised multiple sound sources localization. Two of the methods also perform sound source separation based on binary masking. We start with an overview of previous works in sound source separation (Section 5.1), and present the binary-masking approach in more details in Section 5.2. We then propose *mixed* extensions of the PPCM (Section 4.2) and iPPAM (Section 3.3) methods. These extensions allow to jointly localize and separate multiple sound sources from mixed binaural inputs. Section 5.3 presents the *mixed PPCM* (mPPCM) model, and inference of the model is devised using an expectation-maximization (EM) algorithm referred to as PCESSL. Section 5.4 presents the *mixed PPAM* (mPPAM) model, and inference of the model is solved through a variational EM algorithm referred to as VESSL. Similarly to all current methods in the literature, both PCESSL and VESSL localize multiple sound sources based on the so-called *W-disjoint orthogonality* (WDO) assumption, *i.e.*, one source is strongly dominating at each time-frequency point. In Section 5.5, we propose a radically different approach that does not assume WDO, and is able to localize pair of sources even when a very strong overlap exist, *e.g.*, mixture of two white noise signals. The three proposed methods are thoroughly tested through different experiments, and compared to a state-of-the-art binaural sound source separation and localization technique.

5.1 Previous Work

The problem of sound source separation has been thoroughly studied in the last decades. The literature on the subject is immense, notably because several instances of this problem may be considered, involving different hypothesis. Is there one or several microphones? Are the mixtures synthesized, recorded in a studio, or in an unconstrained environment?

Are signals affected by reverberations? How many sources are there? Are the sources diffuse or spatially narrow? Is the mixture affected by some post-processing effects such as compression or equalization? Can we assume that the sound propagates in a direct path to microphones or are there filtering effects? Are the emitted signals general or do they belong to a specific class?...

Several interesting approaches have been proposed. Some methods achieve separation with a single microphone, based on known acoustic properties of speech signals [Roweis 00, Radfar 07, Bensaid 10], and are therefore limited to speech. Some single microphone methods exploit the redundancy and sparsity in signals such as speech and music to perform the separation based on dictionary-learning [Schmidt 06, Smaragdis 09]. More commonly, separation is done using several microphones. Most techniques are based on a convolutive mixing model expressed in the time-frequency (TF) domain. In each time-frequency bin, the complex recorded Fourier coefficients are expressed as a linear transformation of the emitted Fourier coefficients, possibly perturbed by noises that are independent across microphones. The transformations of individual source coefficients in microphones are called *source images*. Within this framework, the *beamforming* approach consist in estimating an optimal spatial filters to extract a desired signal from the mixture [Cardoso 93, Parra 02, Markovich 09]. This filter act as a linear combination of microphone inputs at each frequency, designed to enhance the target source while attenuating the undesired ones. A common assumption is that the sources are sparse in the TF plane. For instance, binary masking, originally introduced in [Yilmaz 04], extracts one predominant source in each TF bin [Viste 03, Harding 06, Mouba 06, Keyrouz 07, Mandel 10, Woodruff 12]. This approach is described in more detailed in section 5.2 of this chapter. Other techniques known as l_1 -norm minimization relax the assumption that a single source is emitting at each TF point and extracts up to I sources per TF bin, where I is the number of microphones [Bofill 03, Winter 07]. Another recent approach consists in modeling the source images with complex Gaussian random variables whose covariances are parameterized by the spectro-temporal power of the sources [Févotte 05, Duong 09, Ozerov 10]. The model parameters can be estimated in the maximum-likelihood sense. Within this framework, the convolutive mixing model has been recently extended to full-rank covariance matrices [Duong 10]. The generality of this model yields good performance even in reverberant conditions. However, complex Gaussian approaches usually require the estimation of a very large number of parameters, which makes them hard to initialize. Hence, they usually rely on prior-knowledge on the sources, *i.e.*, some of the parameters are already known [Duong 09], or on other source separation algorithms for initialization [Duong 10]. Their efficiency is thus governed by the quality of the initializing algorithms. Another category of methods relies on independent component analysis (ICA) [Comon 10]. ICA usually assumes that signals emitted by sources are non-stationary or non-Gaussian and statistically independent from each other. Since ICA performs poorly in the time-domain in the case of audio mixtures, it is usually applied separately in each frequency subband [Smaragdis 97]. A problematic consequence of this approach, which also occurs in most of the above mentioned techniques [Smaragdis 97, Winter 07, Duong 09, Ozerov 10, Duong 09], is that the source labels are unknown in each frequency subband. Hence, to recover separated signals, one

needs to face the well-known *permutation alignment* problem, *i.e.*, align the source labels across frequencies. This cannot be solved without using prior-knowledge about the source and/or the mixing filters. In contrast with this limitation, an advantage of binary masking is that it can be easily combined with localization-based clustering, as done in [Viste 03, Yilmaz 04, Harding 06, Mouba 06, Keyrouz 07, Mandel 10, Woodruff 12]. Since the localization of each source is estimated, the permutation problem does not occur: The assignment of TF bins to sources is linked to a specific location, which removes ambiguities.

Interestingly, some recent works proposed to assist sound source separation using visual information. In [Wang 05, Khan 13], prior information on source position is included by using face detection in images. Alternatively, [Kidron 05] separate music instruments based on canonical correlation analysis between audio and visual data.

5.2 Binary Masking

In this thesis, we focus on localization-based sound source separation, since we are both interested in the source positions and emitted signals. The setting considered is that of a binaural head placed in a real, unconstrained room. As seen in previous chapter, for a single spatially-narrow emitter, ILD and IPD values depend on the emitter's spatial location, particularly its two dimensional direction vector relative to the head coordinate system. Interaural cues could hence be used for single sound source localization. Matters are more complicated when different sound sources are simultaneously active, from multiple directions. Indeed, the sources mix at each microphone and interaural features not only depend on the directions but also on the relative power spectral density of all sources. The sources' spectra are unknown, challenging analysis of measured interaural features. A common approximation made to simplify this analysis is to assume that at any time-frequency (TF) point that has significant acoustic power, this power is dominated by just a single source. This assumption is referred to as *W-disjoint orthogonality* (WDO) [Yilmaz 04]: One time-frequency point is associated to one of the emitting source only. WDO has been shown to be valid to some extent in the case of mixtures of speech signals [Yilmaz 04]. It was notably successfully applied to the binaural multiple sound sources localization problem we are interested in [Roman 03, Mandel 10, Lee 10, Woodruff 12].

This assumption leads to *binary-masking* sound source separation techniques. The principle of binary-masking is to represent the input signal in the time-frequency domain using short-time Fourier transform (STFT). Points corresponding to the target source are then weighted with 1 and otherwise with 0, thus forming a *binary mask* for each source. Recorded spectrograms are then multiplied by each binary mask and finally converted back to time domain using inverse STFT to obtain separated signals. The pipeline of this method is illustrated in Figure 5.1.

A number of methods combined binary-masking with localization-based clustering for sound source separation, *e.g.*, [Yilmaz 04], [Mouba 06], [Mandel 10]. For example, [Mandel 10] proposed a probabilistic model for multiple sound source localization and

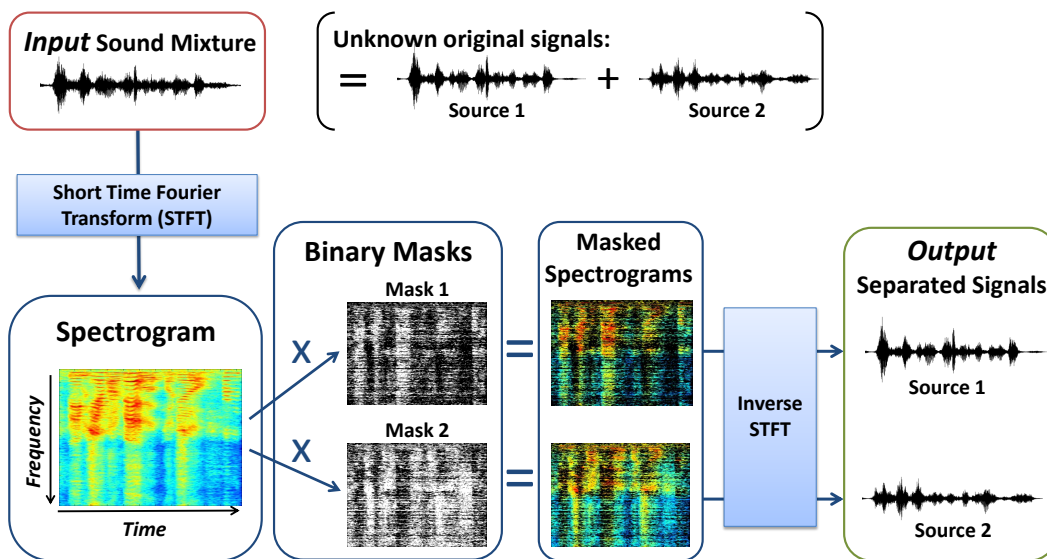


Figure 5.1: Pipeline of the binary-masking approach.

separation based on interaural spatial cues and binary masking. For each sound source, a binary mask and a discrete distribution over interaural time delays is provided. This can be used to approximate the frontal azimuth angle of the sound source using a direct-path sound propagation model, if the distance between the microphones is known.

In section 5.3 and 5.4 we show how, based on WDO, the two single sound source localization models and algorithms proposed in chapter 4, namely PPCM and iPPAM, can be extended to multiple sound sources localization and separation. This is done by modeling the sources binary masks with hidden random variables. These extensions lead to closed-form EM algorithms for parameters estimation, that alternate iteratively between source localization and binary-masking until convergence.

The training data are the same as in chapter 4, *i.e.* a set of N associated source positions and interaural feature vectors $\mathcal{T} = \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^N$. In the case of PPAM, this set is used to learn an optimal set of parameters $\tilde{\theta}$ corresponding to K piecewise affine transformations. The input data are also the same as is chapter 4, *i.e.* a time series of interaural feature vectors with missing values, denoted $\mathcal{S} = \{\{y'_{dt}\}_{d=1,t=1}^{D,T}, \mathcal{X}\}$. This time however, we will assume that M static sound sources are emitting, and that each non-missing interaural value y'_{dt} relates to one of the M sources, *i.e.*, the WDO assumption. The unknown source positions are denoted $\{\mathbf{x}_m\}_{m=1}^M$. The assignment of spectrogram points to sources will be modeled by hidden variables $\mathbf{U} = \{U_{dt}\}_{d=1,t=1}^{D,T}$ such that $U_{dt} = m$ means that y'_{dt} was emitted by source m . With these notations, the binary mask of source m is thus defined by boolean variables $\{(U_{dt} == m)\}_{d=1,t=1}^{D,T}$. \mathbf{U} will hence be referred to as the *masking variables*. Estimating the number of emitting sound sources is an interesting and difficult problem. However, this is left for future work and we assume that the number of emitting sources M is known in the remainder of this chapter.

5.3 Mixed Probabilistic Piecewise Constant Mapping

5.3.1 The mPPCM Model

We extend here to mixture of sound sources the probabilistic piecewise constant mapping (PPCM) model presented in section 4.2 to mixture of sound sources. The resulting model will be called *mixed PPCM* (mPPCM). The problem of simultaneous localization and separation amounts to estimate the masking variables \mathbf{U} and the locations $\{\mathbf{x}_m\}_{m=1}^M$ given an input spectrogram \mathcal{S} . The PPCM model defined in equation (4.4) can be straightforwardly extended to several sources by writing for all d, t such that $\chi_{dt} = 1$:

$$p(y'_{dt}|U_{dt} = m; \boldsymbol{\psi}_{\text{mix}}) = \angle\mathcal{N}(y'_{dt}; g_d(\mathbf{x}_m), \rho_d^2) \quad (5.1)$$

where $\angle\mathcal{N}$ is defined in 4.5 and $\boldsymbol{\psi}_{\text{mix}}$ denotes mPPCM's parameters. To complete the generative model, we define the following prior probability on masking variables:

$$p(U_{dt} = m; \boldsymbol{\psi}_{\text{mix}}) = \omega_{dm} \quad (5.2)$$

where $\omega_{dm} \in [0, 1]$ and $\sum_{m=1}^M \omega_{dm} = 1$ for all $d \in [1 : D]$. Parameters $\{\omega_{dm}\}_{d=1, m=1}^{D, M}$ are constrained so that $\omega_{dm} = \omega_{d'm}$ if the features indexed d and d' correspond to the same frequency channel f in the input spectrogram, *i.e.*, d is the ILD value at f and d' is the IPD value at f . Hence ω_{dm} can be viewed as the *weight* of source m in a given frequency channel, *i.e.*, the proportion of points emitted by m at that frequency with respect to the other sources. In summary, the model parameters are

$$\boldsymbol{\psi}_{\text{mix}} = \{ \{ \rho_{dm}^2, \omega_{dm} \}_{d=1}^D, \mathbf{x}_m \}_{m=1}^M. \quad (5.3)$$

As in section 4.2, we assume that all the spectrogram observations are iid, yielding the following expression for the observed-data log-likelihood of mPPCM:

$$\mathcal{L}_{\text{mPPCM}}(\mathcal{S}; \boldsymbol{\psi}_{\text{mix}}) = \log p(\mathcal{S}; \boldsymbol{\psi}_{\text{mix}}) \quad (5.4)$$

$$= \sum_{\chi_{dt}=1} \log p(y'_{dt}; \boldsymbol{\psi}_{\text{mix}}) \quad (5.5)$$

$$= \sum_{\chi_{dt}=1} \log \left(\sum_{m=1}^M \angle\mathcal{N}(y'_{dt}; g_d(\mathbf{x}_m), \rho_d^2) \right) \quad (5.6)$$

where $\sum_{\chi_{dt}=1} \{.\}$ denotes a sum over all $d \in [1 : D]$ and $t \in [1 : T]$ such that $\chi_{dt} = 1$.

5.3.2 PCESSL: an EM algorithm for mPPCM

Finding optimal source positions amounts to maximize the observed-data log-likelihood $\mathcal{L}_{\text{mPPCM}}$ (5.4) with respect to parameters $\boldsymbol{\psi}_{\text{mix}}$. Once parameters $\tilde{\boldsymbol{\psi}}_{\text{mix}}$ are estimated, one can use the *maximum a posteriori* (MAP) values of masking variables \mathbf{U} to obtain a

binary mask for each source and separate the signals. However, the complexity of expression (5.6), *i.e.* a sum of log-sums of exponentials, does not allow to maximize it in closed-form. An iterative maximization approach is therefore required. We address this maximum-likelihood with missing-data problem within the framework of expectation-maximization (EM). In our case, the E-step computes the posterior probabilities of assigning each spectrogram point to a sound source m (*separation step*), while the M-step maximizes the expected complete-data log-likelihood with respect to the model parameters $\boldsymbol{\psi}_{\text{mix}}$ including source positions $\{\boldsymbol{x}_m\}_{m=1}^M$ (*localization step*).

The expected complete-data log-likelihood writes:

$$Q(\boldsymbol{\psi}_{\text{mix}}|\boldsymbol{\psi}_{\text{mix}}^{(l-1)}) = \sum_{d=1, t=1, m=1}^{D, T, M} r_{dtm}^{(i)} \log \omega_{dm} p(y'_{dt}|U_{dt} = m; \boldsymbol{\psi}_{\text{mix}}^{(l-1)}) \quad (5.7)$$

where $^{(i)}$ denotes the i -th iteration and $r_{dtm}^{(i)}$ the posterior probability $p(U_{dt} = m|\mathcal{S}; \boldsymbol{\psi}_{\text{mix}}^{(l-1)})$. Posterior probabilities are updated in the *E-step* as follows:

$$\begin{cases} r_{dtm}^{(i)} = \frac{\omega_{dm} p(y'_{dt}|U_{dt} = m; \boldsymbol{\psi}_{\text{mix}}^{(l-1)})}{\sum_{i=1}^M \omega_{di} p(y'_{dt}|U_{dt} = i; \boldsymbol{\psi}_{\text{mix}}^{(l-1)})} & \text{when } \chi_{dt} = 1, \\ r_{dtm}^{(i)} = 0 & \text{when } \chi_{dt} = 0. \end{cases} \quad (5.8)$$

The *M-step* maximizes (5.7) with respect to $\boldsymbol{\psi}_{\text{mix}}$. By combining (5.1) with (5.7), the equivalent criterion to minimize for each m writes:

$$\sum_{d=1, t=1}^{D, T} r_{dtm}^{(i)} \left(\log \left(\frac{\rho_{dm}^2}{\omega_{dm}} \right) + \frac{\Delta(y'_{dt}, g_d(\boldsymbol{x}_m))^2}{\rho_{dm}^2} \right). \quad (5.9)$$

This can be differentiated with respect to $\{\omega_{dm}, \rho_{dm}^2\}_{d=1}^D$ to obtain closed-form expressions for the updated parameters $\{\omega_{dm}^{(i)}, \rho_{dm}^{2(i)}\}_{d=1}^D$ as a function of $\boldsymbol{x}_m^{(i)}$:

$$\omega_{dm}^{(i)} = \frac{\bar{r}_{dm}}{\bar{\chi}_d}, \quad \text{with } \bar{r}_{dm} = \sum_{t=1}^T r_{dtm}^{(i)} \quad \text{and} \quad \bar{\chi}_d = \sum_{t=1}^T \chi_{dt} \quad (5.10)$$

$$\rho_{dm}^{2(i)} = \frac{1}{\bar{r}_{dm}} \sum_{t=1}^T r_{dtm}^{(i)} \Delta(y'_{dt}, g_d(\boldsymbol{x}_m^{(i)}))^2 \quad (5.11)$$

To account for the equality constraint between source weights corresponding to the same frequency channel, these weights are set to their mean value. For example, to account for the constraint $\omega_{dm} = \omega_{d'm}$ we set $\omega_{dm}^{(i)} = \omega_{d'm}^{(i)} = (\omega_{dm}^{(i)} + \omega_{d'm}^{(i)})/2$. By substituting (5.10) and (5.11) into (5.9) the position update $\boldsymbol{x}_m^{(i)}$ is obtained by minimizing the following expression with respect to \boldsymbol{x}_m :

$$\sum_{d=1} \bar{r}_{dm} \log \left(1 + \frac{\Delta(\bar{y}'_d, g_d(\boldsymbol{x}_m))^2}{1/\bar{r}_{dm} \sum_{t=1}^T r_{dtm}^{(i)} \Delta(y'_{dt}, \bar{y}'_d)^2} \right) \quad \text{where } \bar{y}'_d = \frac{1}{\bar{r}_{dm}} \sum_{t=1}^T r_{dtm}^{(i)} y'_{dt}. \quad (5.12)$$

This expression reminds us the single-source localization criterion (4.8) where χ_{dt} has been replaced by $r_{dtm}^{(i)}$. It is evaluated for all position $\{\mathbf{x}_n\}_{n=1}^N$ in the training set \mathcal{T} , and $\mathbf{x}_m^{(i)}$ is set to the position $\mathbf{x}_{\hat{n}}$ minimizing (5.12). Then, $\mathbf{y}_{\hat{n}} = g(\mathbf{x}_{\hat{n}})$ is substituted back in (5.11) to obtain $\rho_{dm}^{2(i)}$. This is repeated for each source m . The E- and M-steps are iterated until convergence of the observed-data log-likelihood $\mathcal{L}_{\text{mPPCM}}$ (5.4). This algorithm is referred to as PCESSL for *piecewise constant EM for sound source separation and localization*. PC may also stand for *pointwise-constrained*, since the means of interaural features are constrained by the points in the training set \mathcal{T} for each source.

5.3.3 PCESSL's initialization strategies

An EM procedure is guaranteed to converge to a local maximum of the observed-data log-likelihood (5.4). However, the non-injectivity nature of the position-to-interaural-feature mapping function g_d and the high cardinality of ψ_{mix} leads to a very large number of such local maxima, especially when the size N of the training set is large. This makes our algorithm very sensitive to initialization. One way to avoid being trapped in local maxima is to initialize the mixture's parameters at random several times. This cannot be easily applied here since there is no straightforward way to initialize the model's variances. Alternatively, one may randomly initialize the masking variables U and then proceed with the M-step. However, extensive simulated experiments revealed that this solution fails to converge to the ground-truth solution in most of the cases. We therefore propose to combine these strategies by randomly perturbing both the source locations and the source assignments during the first stages of the algorithm. We developed a *stochastic initialization* procedure similar in spirit to SEM [Celeux 92]. We note that exploiting stochasticity to escape from local maxima is a commonly used principle in global optimization [Zhigljavsky 08]. The SEM algorithm includes a stochastic step (S) between the E- and the M-step, during which random samples $r_{dtm}^* \in \{0, 1\}$ are drawn from the posterior probabilities (5.8). These samples are then used instead of (5.8) during the M-step. To initialize our algorithm, we first set $r_{dtm}^{(0)} = 1/M$ for all m and then proceed through the sequence S M* E S M, where M* is a variation of M in which the source positions are drawn randomly from \mathcal{T} instead of minimizing (5.12). In practice, ten such initializations are used to enforce algorithm convergence, and only the one providing the best log-likelihood after two iterations is iterated fifteen more times.

A second technique was used to overcome local maxima issues due to the large number of parameters. During the first ten steps of the algorithm only, a unique pair of noise variances $(\rho_m^{2(\text{ILD})}, \rho_m^{2(\text{IPD})})$ is estimated for each source m instead of D variances $\{\rho_{dm}^2\}_{d=1}^D$. This is done by calculating the means of variance updates (5.11) over ILD and IPD indexes, weighted by \bar{r}_{dm} . Note that these means depend on the unknown optimal source position $\mathbf{x}_m^{(i)}$. The optimal position $\mathbf{x}_m^{(i)}$ is actually the one minimizing $\rho_m^{2(\text{ILD})} \rho_m^{2(\text{IPD})}$. The latter is thus calculated for all positions in \mathcal{T} and only the minimum is kept. Intensive experiments showed that the proposed initialization methods converge to a global optimum in most of the cases.

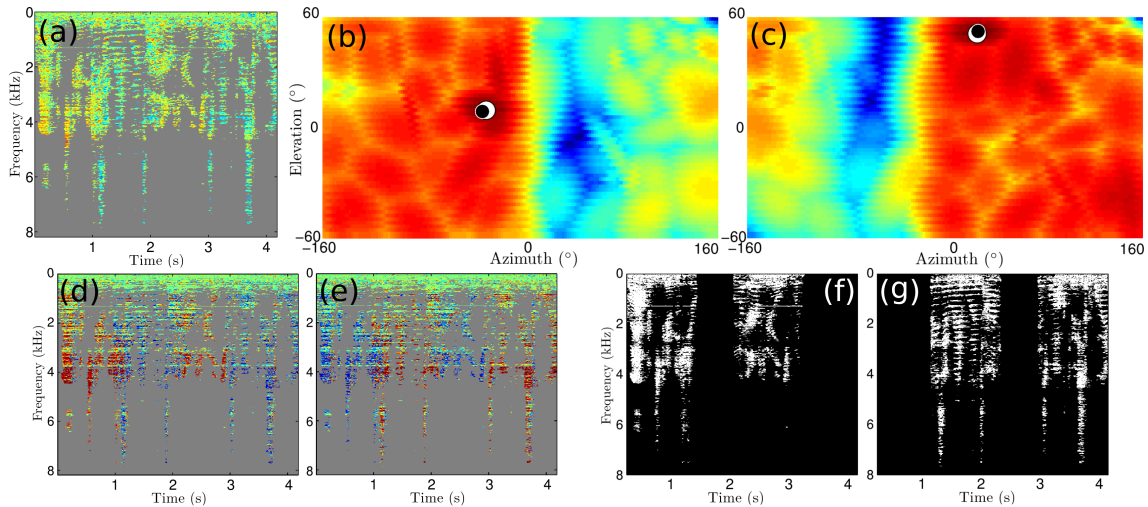


Figure 5.2: Examples of input and output for VESSL using the audio-motor dataset. (a) Input mixed ILD spectrogram. (b,c) Output log-density of each source position as determined by $q_{X,Z}^{(\infty)}$. Ground-truth source positions are noted with a black dot, and the peak of the log-density with a white circle. (d,e) Output masking variables probabilities $q_U^{(\infty)}$. (f,g) Ground truth binary masks. Red color denotes high values, blue color low values, and grey colors missing observations.

5.4 Probabilistic Piecewise Affine Inversion in Mixtures

We now extend the single sound source localization method iPPAM described in section 4.3 to multiple sound source separation and localization. In the GLLiM framework, this corresponds to a piecewise affine inversion problem, where observed signals generated from multiple source positions (modeled as latent variables) are both mixed and corrupted by noise. We extend the PPAM model presented in section 3.3 to this more general case. The resulting model will be called *mixed PPAM* (mPPAM). We propose a general variational expectation-maximization (VEM) framework [Beal 03] to solve for mPPAM inference. The VEM algorithm described below will be referred to as *variational EM for sound source separation and localization* (VESSL). Typical examples of the algorithm’s input and output are shown in Figure 5.2.

5.4.1 The mixed PPAM model

Let $\tilde{\theta}$ be a set of learned PPAM parameters. Given a time series of T noisy interaural feature vectors $\mathcal{S} = \{\{y_{dt}'\}_{d=1,t=1}^{D,T}, \mathcal{X}\}$, we are looking for the M emitting sound source positions, denoted by $\{\mathbf{x}_m\}_{m=1}^M \subset \mathbb{R}^L$. In the GLLiM framework, source positions will be treated as latent random variable $\{\mathbf{X}_m\}_{m=1}^M$ whereas they were treated as parameters in the mixed PPCM models. To make the notations more compact, we will use $\mathbf{x} = \{\mathbf{x}_m\}_{m=1}^M \subset \mathbb{R}^L$ and $\mathbf{X} = \{\mathbf{X}_m\}_{m=1}^M$ in this section. To deal with mixed data, we also use the masking variable $\mathbf{U} = \{U_{dt}\}_{d=1,t=1}^{D,T}$ introduced in section 5.2. We assume that each source m is associated to one affine transformation denoted by the hidden variable

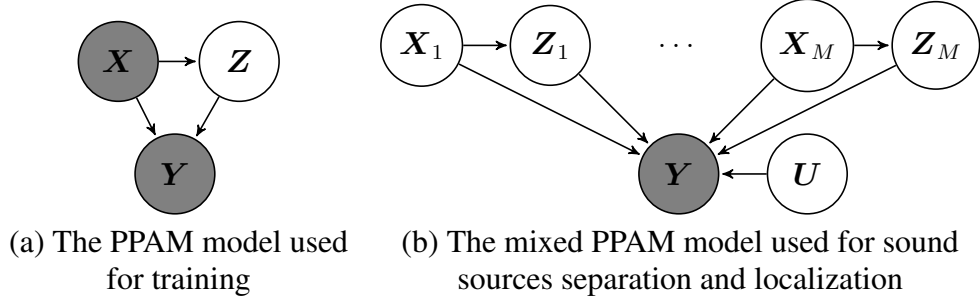


Figure 5.3: Graphical models of PPAM and mixed PPAM. White means unobserved, gray means observed.

$Z_m \in [1 : K]$. The set of affine transformation assignments is denoted $\mathbf{Z} = \{Z_m\}_{m=1}^M$. The only observed data are interaural cues \mathcal{S} while all the other variables, namely masking variables $\mathbf{U} \in \mathcal{U}$, source positions $\mathbf{X} \subset \mathcal{X}$ and affine transformation assignments $\mathbf{Z} \in \mathcal{Z}$ are hidden. Based on these extensions, the observation model of mPPAM with equal diagonal noise covariance matrices (4.10) writes

$$p(\mathcal{S}|\mathbf{U}, \mathbf{X}, \mathbf{Z}; \boldsymbol{\theta}_{\text{mix}}) = \prod_{\chi_{dt}=1} p(y'_{dt}|u_{dt}, \mathbf{x}_{u_{dt}}, z_{u_{dt}}) \quad (5.13)$$

$$= \prod_{\chi_{dt}=1} \mathcal{N}(y'_{dt}; \tilde{\mathbf{a}}_{dk}^\top \mathbf{x}_m + \tilde{b}_{dk}, \sigma_d^2). \quad (5.14)$$

where $\boldsymbol{\theta}_{\text{mix}}$ denotes mPPAM's parameters. Note that $\boldsymbol{\theta}_{\text{mix}}$ includes the learned PPAM parameters $\tilde{\boldsymbol{\theta}}$. These will remain fixed, except noise variances $\boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, \dots, \sigma_D^2)$ that are re-estimated in order to account for possibly higher noise levels in the mixed observed signals compared to training.

We assume that the M source position variables associated to their affine transformations are independent, yielding

$$p(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta}_{\text{mix}}) = \prod_{m=1}^M p(\mathbf{x}_m, z_m; \boldsymbol{\theta}_{\text{mix}}). \quad (5.15)$$

Masking variables are also assumed to be independent over both time and frequency, so that

$$p(\mathbf{u}; \boldsymbol{\theta}_{\text{mix}}) = \prod_{d,t} p(u_{dt}; \boldsymbol{\theta}_{\text{mix}}). \quad (5.16)$$

We define the prior on masking variables by

$$p(U_{dt} = m; \boldsymbol{\theta}_{\text{mix}}) = \lambda_{dm} \quad (5.17)$$

where $\lambda_{dm} \in [0, 1]$ and $\sum_{m=1}^M \lambda_{dm} = 1$ for all $d \in [1 : D]$. Parameters $\boldsymbol{\lambda} = \{\lambda_{dm}\}_{d=1, m=1}^{D, M}$ represent the relative presence of each source in each feature and frequency channel (sources' weights). Finally, masking variables and source positions are assumed independent, so that we get the following hierarchical decomposition of the full model:

$$p(\mathcal{S}, \mathbf{U}, \mathbf{X}, \mathbf{Z}; \boldsymbol{\theta}_{\text{mix}}) = p(\mathcal{S}|\mathbf{U}, \mathbf{X}, \mathbf{Z}; \boldsymbol{\theta}_{\text{mix}})p(\mathbf{X}, \mathbf{Z}; \boldsymbol{\theta}_{\text{mix}})p(\mathbf{U}; \boldsymbol{\theta}_{\text{mix}}). \quad (5.18)$$

The complete set of mPPAM's parameters is

$$\boldsymbol{\theta}_{\text{mix}} = \{\{\tilde{\Gamma}_k, \tilde{\mathbf{c}}_k, \tilde{\mathbf{A}}_k, \tilde{\mathbf{b}}_k\}_{k=1}^K, \boldsymbol{\Sigma}, \boldsymbol{\lambda}\}. \quad (5.19)$$

Note that as in PPAM, affine components' weights $\{\pi_k\}_{k=1}^K$ are supposed equal to $1/K$ and are hence omitted. PPAM and mPPAM's graphical models are showed in Figure 5.3.

Notice that PPAM, where observed position-to-interaural-cue couples $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$ are given for training, can be viewed as a particular instance of mixed PPAM where $T = M = N$ and \mathbf{X} and \mathbf{U} are completely known ($U_{dn} = n$ for all n). Hence, amongst the parameters $\boldsymbol{\theta}_{\text{mix}}$ of mixed PPAM, the values of $\{\tilde{\Gamma}_k, \tilde{\mathbf{c}}_k, \tilde{\mathbf{A}}_k, \tilde{\mathbf{b}}_k\}_{k=1}^K$ have already been estimated during the training stage and can be fixed to these values. Only the parameters $\{\boldsymbol{\Sigma}, \boldsymbol{\lambda}\}$ remains to be estimated, while \mathbf{X} and \mathbf{U} are hidden variables.

5.4.2 The VESSL algorithm

The problem of localizing and separating sound sources amounts to estimate the probability of hidden masking variables \mathbf{U} and hidden source positions \mathbf{X} given observed data \mathcal{S} and some learned PPAM's parameters $\tilde{\boldsymbol{\theta}}$. As in section 5.3, estimation of the parameters is a maximum-likelihood with missing data problem, that can be solved using EM. However, a standard EM procedure would require the estimation of the posterior distribution $p(\mathbf{u}, \mathbf{x}, \mathbf{z} | \mathcal{S}; \boldsymbol{\theta}_{\text{mix}})$ in the E-step, in order to calculate the expected complete-data log-likelihood with respect to this distribution. Unfortunately, it turns out that this cannot be done in closed-form. We therefore developed an approximate inference of the parameters through a *variational expectation-maximization* (VEM) procedure. Denoting current parameter values by $\boldsymbol{\theta}_{\text{mix}}^{(i)}$, the proposed VEM algorithm provides, at each iteration (i), an approximation $q^{(i)}(\mathbf{u}, \mathbf{x}, \mathbf{z})$ of the posterior probability $p(\mathbf{u}, \mathbf{x}, \mathbf{z} | \mathcal{S}; \boldsymbol{\theta}_{\text{mix}}^{(i)})$ that factorizes as

$$q^{(i)}(\mathbf{u}, \mathbf{x}, \mathbf{z}) = q_U^{(i)}(\mathbf{u}) q_{X,Z}^{(i)}(\mathbf{x}, \mathbf{z}) \quad (5.20)$$

where $q_U^{(i)}$ and $q_{X,Z}^{(i)}$ are probability distributions on \mathcal{U} and $\mathcal{X} \times \mathcal{Z}$ respectively. Such a factorization may seem drastic but its main beneficial effect is to replace potentially complex stochastic dependencies between latent variables with deterministic dependencies between relevant moments of the two sets of variables. It follows that the E-step becomes an approximate E-step that can be further decomposed into two sub-steps whose goal is to update $q_{X,Z}$ and q_U in turn. The algorithm's updating rules are:

$$\mathbf{E-XZ \ step:} \quad q_{X,Z}^{(i)}(\mathbf{x}, \mathbf{z}) \propto \exp \mathbb{E} q_U^{(i-1)}[\log p(\mathbf{x}, \mathbf{z} | \mathcal{S}, \mathbf{u}; \boldsymbol{\theta}_{\text{mix}}^{(i)})] \quad (5.21)$$

$$\mathbf{E-U \ step:} \quad q_U^{(i)}(\mathbf{u}) \propto \exp \mathbb{E} q_{X,Z}^{(i)}[\log p(\mathbf{u} | \mathcal{S}, \mathbf{x}, \mathbf{z}; \boldsymbol{\theta}_{\text{mix}}^{(i)})] \quad (5.22)$$

$$\mathbf{M \ step:} \quad \boldsymbol{\theta}_{\text{mix}}^{(i+1)} = \operatorname{argmax}_{\boldsymbol{\theta}_{\text{mix}}} \mathbb{E} q_U^{(i)} q_{X,Z}^{(i)}[\log p(\mathcal{S}, \mathbf{u}, \mathbf{x}, \mathbf{z}; \boldsymbol{\theta}_{\text{mix}})]. \quad (5.23)$$

The E-XZ step may be view as the localization step, since it amounts to estimate a probability distribution over the source position space for each source. The E-U step can be see as the separation step since it amounts to estimate a probability distribution over the

space of binary masks. Detailed derivations of closed-form expressions for all these steps as well as the method used to check convergence of the algorithm are given in appendix 5.A.

5.4.3 VESSL's initialization strategies

Extensive experiments have shown that VESSL's objective function, *i.e.*, the variational free energy (5.43), had a large number of local maxima using real world sound mixtures. This may be due to the combinatorial sizes of the set of all possible binary masks \mathcal{U} and the set of all possible affine transformation assignments \mathcal{Z} . Indeed, the procedure has shown to be more sensitive to initialization and to get trapped in suboptimal solutions more often as the size of the spectrogram and the number of transformation K increased. On the other hand, too few local affine transformations K make the mapping very imprecise. We thus developed a novel efficient way to deal with the well established local maxima problem, referred to as *multi-scale initialization*. The idea is to train PPAM at different *scales*, *i.e.*, with a different number of transformation K each time, yielding to different sets of trained parameters $\tilde{\theta}_K$ where, *e.g.*, $K = 1, 2, 4, 8 \dots, 64$. When proceeding to the inverse mapping, we first run VESSL from a random positions initialization using $\tilde{\theta}_1$. We then use the obtained masks and positions to initialize a new VEM algorithm using $\tilde{\theta}_2$, then $\tilde{\theta}_4$, and so on so forth until the desired value for K .

To further improve the convergence of each *sub-scale* algorithm an additional constraint was added, referred to as *progressive masking*. During the first iteration, the mask of each source is constrained such that at each time window t , all the frequency bins of interaural vector \mathbf{y}'_t are assigned to the same source. This is done by adding a product over t in the expression of $q_{U_{dt}}^{(1)}(m)$ (5.39). Similarly to what is done in [Mandel 10], this constraint is then progressively released at each iteration by dividing time windows into 2,4,8... frequency blocks until the total number of frequency bins is reached. Combining these two strategies dramatically increased the algorithm's performance.

5.4.4 VESSL's termination

Once VESSL has converged to an optimal set of parameters $\theta_{\text{mix}}^{(\infty)}$ and optimal posterior distributions $q_{X,Z}^{(\infty)}(\mathbf{x}, \mathbf{z})$ and $q_U^{(\infty)}(\mathbf{u})$, we can maximize these distributions with respect to \mathbf{x} , \mathbf{z} and \mathbf{u} to obtain *maximum a posteriori* (MAP) estimates. These estimates yield optimal positions and binary masks for all sources. We have:

$$(\mathbf{x}_m^{\text{MAP}}, z_m^{\text{MAP}}) = (\boldsymbol{\mu}_{km}^{(\infty)}, \hat{k}) \quad \text{with } \hat{k} = \underset{k}{\operatorname{argmax}} \nu_{km}^{(\infty)} \quad (5.24)$$

$$\text{and } U_{dt}^{\text{MAP}} = \underset{m}{\operatorname{argmax}} q_{U_{dt}}^{(\infty)}(m) \quad (5.25)$$

where $\boldsymbol{\mu}_{km}$ and ν_{km} are respectively defined in (5.32) and (5.34). Note that as shown in Figure 5.2(b...e), the algorithm not only provides MAP estimates, but also complete

posterior distributions over both the 2D space of sound source positions \mathcal{X} and the space of masking variables \mathcal{U} . Obtaining probability distributions over those space may be very useful. For example, one could combine VESSL’s output with probabilistic knowledge obtained from other sensory modalities such as vision. It also provides some higher order information about the source location. For example, the distributions in figures 5.2(b) and 5.2(c) suggest that one source is located almost certainly on the left, while the other is located almost certainly on the right.

5.5 Co-Localization of Sound Source Pairs

As explained in section 5.2, both mPPCM-EM and VESSL strongly rely on the WDO assumption, *i.e.*, they assume that only one source is emitting at a given time-frequency (TF) point. In fact, to the best of our knowledge, this is also the case for all current multiple SSL techniques, *e.g.* [Aarabi 02, Roman 03, Yilmaz 04, Mandel 10, Lee 10, Woodruff 12] to name just a few. Indeed, a common point of these techniques is to perform some form of clustering in the TF plane before localizing individual sources based on the interaural cues in each cluster. Some perform the clustering by selecting peaks in histograms of ITDs accumulated over frequency channels [Aarabi 02, Yilmaz 04, Lee 10]. Others, including [Mandel 10], mPPCM-EM and VESSL, rely on expectation-maximization (EM) inference which is more computationally demanding. In [Roman 03, Woodruff 12], the WDO assumption allows to define a basic link between interaural cues and reference source azimuth. It is then combined with a statistical model of ILD/ITD distribution that takes into account interfering sources, reverberation or background noise. Although WDO has been shown to be valid to some extent in the case of mixtures of speech signals [Yilmaz 04], it has limitations when dealing with interaural cues. Indeed, these cues are strongly perturbed when two sources are emitting in the same TF point, even if one dominates the other, and this often results in localization errors.

Based on our probabilistic model PPAM, we propose a radically different approach, that does not explicitly assume WDO, and requires no clustering. Rather, we show that binaural cues resulting from a mixture of *two sources emitting simultaneously* ($M = 2$) can surprisingly be directly mapped onto *a pair of 2D direction vectors*, each vector corresponding to a source location. To achieve this, we simply push further the concept of acoustic space mapping, beyond the single source case.

Using the audio-visual single white-noise sound source training sets presented in section 2.3.3, we computationally built a two-source training set, that associates recordings of two simultaneously emitting white-noise sources to their pair of 2D directions. Source-pair recordings are simply emulated by summing a pair of white-noise binaural recordings in the training set, each corresponding to an individual source location. We can then compute mean interaural feature vectors from these recordings, as explained in section 2.2. However, this time, each one of these vector is associated to a 4-dimensional vector corresponding to the pair of 2D directions ($L = 4$). Since such data were obtained from the single source training set, note that acquiring *mixed training data* adds no complexity. Due to the randomness of white-noise spectrograms’ spectral density, the spectral density

ratio of the two sources is different at each point. The training set thus capture mean interaural cues perceived for a given pair of position, when different *spectral density ratios* exist between the two sources, and the WDO is not assumed.

From mixed training data, a set of PPAM parameters $\tilde{\theta}$ can be learned. Given a new binaural recording of a 2-source mixture, the learned parameters $\tilde{\theta}$ can then be used to recover the 4D directions pair using iPPAM’s spectrogram inversion formula (4.17). We will refer to the resulting source-pair localization technique as *co-localization* (CoL). Since CoL does not make any assumption on the TF overlap between sources, it has the potential to deal with situations where a very large overlap exists, which cannot be handled by WDO-based methods.

5.6 Results

5.6.1 Tested methods

We evaluated the performance of the two proposed sound sources separation and localization (SSSL) techniques PCESSL and VESSL, as well as the source-pair localization method CoL. PCESSL and VESSL’s separation results are compared to the state-of-the-art SSSL method MESSL, also based on EM and binary masking. The version MESSL-G used includes a garbage component and ILD priors to better account for reverberations and is reported to outperform four methods in reverberant conditions in terms of separation [Yilmaz 04, Buchner 05, Mouba 06, Sawada 07]. PCESSL, VESSL and CoL’s localization results are compared to those of MESSL-G and PHAT histogram. MESSL-G and PHAT do not rely on a training set to localize sources. Rather, they estimate a time difference of arrival for each source, which can be mapped to a one-dimensional azimuth value, using a linear regressor. Since TDOAs induce front-back localization ambiguities, MESSL-G and PHAT were tested on mixtures containing frontal sources only (azimuth between -90° and 90°).

We evaluated sound separation performance using the standard metrics Signal to Distortion Ratio (SDR) and Signal to Interferer Ratio (SIR) introduced in [Vincent 06]. SDR and SIR scores of tested methods were also compared to those obtained with the ground truth binary masks or *oracle masks* [Yilmaz 04] and to those of the original mixture. The oracle mask of a source is set to 1 at every spectrogram point in which the source is at least as loud as the combined other sources and 0 everywhere else. Oracle masks provide an upper bound for binary masking methods. This bound cannot be reached in practice because it requires the knowledge of the original signals. Conversely, the mixture scores provide a lower bound, as no mask is applied.

5.6.2 Multiple sound sources separation and localization

In a first experiment we tested and compared the performance of WDO-based source separation methods, namely PCESSL, VESSL, MESSL and PHAT. All methods were

evaluated both on the audio-motor dataset (section 2.3.2) and the audio-visual dataset (section 2.3.3). Input data consisted in mixtures of 2 or 3 speech sources. Mixtures were obtained by summing up test recordings from the datasets. Test recordings were cut to last 2 seconds. The mixtures were built so that at least 2 sources were emitting at the same time in 2/3 of the input signal. This creates quite challenging mixtures, while keeping the WDO assumption realistic.

Table 5.1(a) shows localization and separation results obtained with all the methods on the audio-motor dataset. As in section 4.4, the complete training set was cut to have a planar rather than cylinder topology and contained 9,600 points. PCESSL and VESSL were trained on a random subset of the complete training set with 9,000 points. The two algorithms were tested on 200 mixture of sound sources emitting from directions outside the training set. VESSL was used with $K = 128$ affine components.

Both PCESSL and VESSL outperform MESSL-G in terms of separation and localization accuracy. Although MESSL-G and PHAT azimuth error are similar to PCESSL and VESSL in the 3 sources case, our methods also provide an accurate estimation of the elevation angle, which cannot be done using MESSL-G or PHAT. Consistently with the single-source localization results of section 4.4, PCESSL does better than VESSL in terms of localization on the audio-motor dataset. This is expected considering the large training set size used in PCESSL ($N = 9,000$) compared to only $K = 128$ affine components used in VESSL. However, note that VESSL yields less localization outliers than PCESSL in the 3 source case. PCESSL provides the best sound source separation performance, with an average of 5 to 6 dB improvement in SDR and 7 to 9dB improvement in SIR with respect to the original mixture.

Table 5.1(b) shows localization and separation results obtained with all the methods on the audio-visual dataset. PCESSL and VESSL were trained on the complete white noise dataset with 432 points, and $K = 32$ was used for VESSL. All the algorithms were tested on 200 mixtures of speech signals from the audio-visual test set. Still consistently with the single-source localization results of section 4.4, VESSL is the algorithm performing best in terms of localization using the audio-visual training set. Although the mean horizontal error of inliers in the 2 sources case is slightly lower using PCESSL, it yielded 4 times more outliers. VESSL yields an average error of around 40 pixels only (around 1.7°) both horizontally and vertically, even in the very challenging case of 3 source mixtures. VESSL also outperform the other algorithms in terms of source separation, with an average of 4 dB improvement in SDR and 3 to 6 dB improvement in SIR with respect to the original mixtures.

Computational times of the different algorithms to process mixtures of 2 or 3 sources are showed in Table 5.2. The computational time of all the EM-based method is far beyond real-time implementation, due to their costly iterative procedure. The slowest method is VESSL, notably because of the complex algebraic computations involving large arrays. This time could be considerably reduced using parallelism, better initialization procedures, and code optimization. For the smaller audio-visual training set ($K = 32$ for VESSL and $N = 432$ for PCESSL), real-time implementations could probably be obtained.

Table 5.1: Comparing the average and standard deviation (Avg±Std) of inlying errors in azimuth (Az) and elevation (El), as well as the Avg±Std of Signal to Distortion Ratio (SDR) and Signal to Inteferer Ratio (SIR) over 200 localization/separation tasks in mixtures of 2 to 3 sources using different methods. The percentage of outliers (Out) is also showed for each method. An estimate is considered outlier if the distance error is more than 45° in the audio-motor datasets, and more than 300 pixels in the audio-visual dataset.

Method used	2 sources				3 sources					
	Az	El	Out	SDR (dB)	SIR (dB)	Az	El	Out	SDR (dB)	SIR (dB)
PCESSL	1.48±2.4	0.96±2.2	0.0	5.63±1.7	9.43±1.7	2.18±4.3	1.80±4.2	18	2.52±2.1	4.28±2.1
VESL	3.26±3.7	2.43±3.7	1.1	4.60±1.8	7.33±1.8	4.73±6.0	2.76±4.0	14	2.07±1.9	2.81±1.9
MESL-G	4.50±7.0		4.0	2.42±1.5	5.92±4.6	9.02±12		12	1.29±1.3	2.72±4.3
PHAT	4.50±1.5		4.0			8.94±12		11		
Mixture				0.01±3.2	0.23±3.1				-3.2±2.3	-2.9±2.2
Oracle				13.0±2.0	21.7±2.0				10.9±1.7	19.3±2.0

(a) **Audio-motor dataset** (Localization errors in degrees)

Method used	2 sources				3 sources					
	Az	El	Out	SDR (dB)	SIR (dB)	Az	El	Out	SDR (dB)	SIR (dB)
PCESSL	22.9±28	35.3±61	14	2.65±2.0	4.31±2.0	57.4±64	66.0±76	12	0.47±1.2	-0.1±1.2
VESL	28.4±31	36.5±48	3.3	3.81±1.7	6.11±1.7	37.4±49	44.7±56	11	0.85±1.4	0.13±1.4
MESL-G	108±89		24	1.79±1.4	3.08±3.6	113±83		21	0.79±0.9	0.07±3.4
PHAT	105±88		25			110±85		21		
Mixture				0.01±2.6	0.19±2.5				-3.2±2.1	-2.9±2.1
Oracle				14.2±2.2	23.0±2.6				12.3±2.2	20.9±2.5

(b) **Audio-visual dataset** (Localization errors in pixels)

Table 5.2: Mean computational times of different methods in seconds to process mixtures of 2 or 3 two-second sources. Times showed are averages over 200 localization tasks.

Method	2 sources	3 sources
PCESSL ($N = 9,000$)	211	287
PCESSL ($N = 432$)	44.0	72.5
VESSL ($K = 128$)	310	542
VESSL ($K = 32$)	90.3	143
MESSL-G	41.6	67.9
PHAT	0.62	0.69

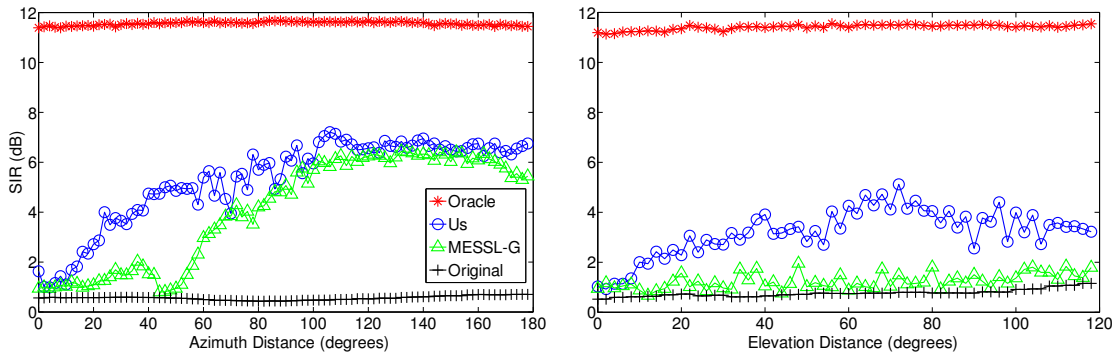


Figure 5.4: SIR as a function of azimuth and elevation separation between two sources in the audio-motor dataset. Left: one source fixed at $(-90^\circ, 0^\circ)$ while the other takes 90 positions between $(-90^\circ, 0^\circ)$ and $(+90^\circ, 0^\circ)$. Right: one source fixed at $(0^\circ, -60^\circ)$ while the other takes 60 positions between $(0^\circ, -60^\circ)$ and $(0^\circ, +60^\circ)$. SIRs are averaged over 6 mixtures of 2 sources (12 targets). Top-to-down: Oracle (*), PCESSL (Us) (○), MESSL-G (△), and original mixture (+).

Figure 5.4 shows SIR separation scores obtained with PCESSL (us), MESSL-G, the oracle mask and the original mixture as a function of the azimuth and elevation spacing between the 2 sources. As can be seen, PCESSL and MESSL perform similarly when the two sources are well separated in azimuth, *i.e.*, more than 90° apart. However, MESSL yields poor results (similar to the original mixture) when sources are nearby in azimuth or share the same azimuthal plane with different elevations (tilt angles). In contrast, PCESSL yields reasonable separation scores even in these cases. This is because MESSL relies on the estimation of a probability density in a discretized TDOA space for each source, and does not account for more subtle spatial cues induced by the HRTF.

5.6.3 Co-localization of overlapping source pairs

In a second experiments, we tested CoL and the other algorithms on 2-source mixtures where the WDO was strongly violated. Algorithms were tested on 1 second mixtures of speech + speech (S+S), speech + white-noise (S+WN) and white-noise + white-noise (WN+WN) signals, where both sources are 100% overlapping in time. These tests were ran on the audio-visual dataset. The methods PCESSL, VESSL, MESSL and PHAT are not supposed to work well in such conditions, since they rely on WDO. On the other hand, CoL is based on a training with strongly overlapping (WN+WN) mixtures, and is

Table 5.3: Localization error average and standard deviation (avg±std) in pixels for different mixture types and different methods. Avgs and stds are calculated over inlying estimates, among 200 one second mixtures of 2 sources. Estimates are considered outliers if their distance to ground-truth is more than 300 pixels. Percentages of outliers are given in columns “out”.

Mixture	WN+WN			WN+S (WN)			WN+S (S)			S+S		
	horizontal	vertical	out	horizontal	vertical	out	horizontal	vertical	out	horizontal	vertical	out
Method												
CoL:ILPD	17.4±19	22.4±24	0.1	18.9±27	15.7±22	0.0	69.0±65	77.3±67	12	31.5±31	44.0±48	1.1
CoL:ILD	23.3±34	24.6±34	0.2	25.4±38	24.5±34	1.6	68.1±60	78.7±68	12	45.5±53	52.2±58	5.9
CoL:IPD	25.9±28	32.5±41	0.6	22.5±26	19.7±28	0.6	76.9±65	87.4±70	14	43.8±43	55.9±57	2.9
PCESSL	34.3±43	45.4±59	6.5	12.1±13	18.2±11	1.0	89.8±77	105±88	33	46.6±59	58.7±76	9.5
VESSL	63.5±68	69.8±72	23	13.0±18	16.2±23	2.0	120±78	106±75	46	71.4±68	76.6±73	17
MESSL-G	82.9±75	—	27	54.5±76	—	28	136±87	—	35	81.7±76	—	22
PHAT	81.2±75	—	27	54.7±76	—	28	133±87	—	35	83.7±76	—	22

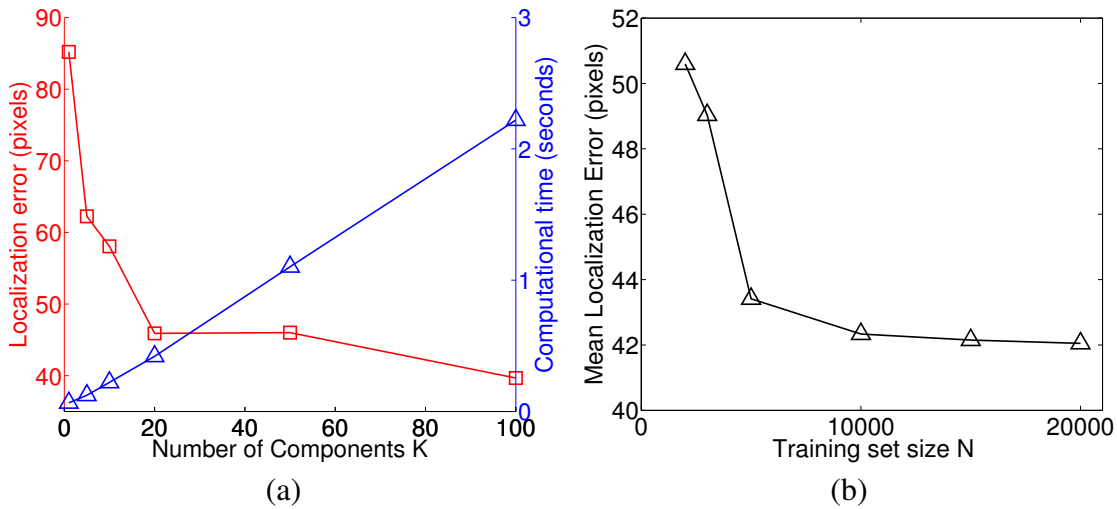


Figure 5.5: (a) Influence of K^* on the mean localization distance error and computational time of a 1 second S+S mixture ($N^* = 20,000$). (b) Influence of N^* on the mean localization distance error of a 1 second S+S mixture ($K^* = 100$).

supposed to perform better. In practice the training dataset of CoL was built by randomly picking $N^* = 20,000$ distinct source-pairs out of $432 \times 433/2 = 93,528$ possible pairs in the audio-visual training set. PPAM was then trained on these data with $K^* = 100$ piecewise affine components. iPPAM's formula (4.17) with $L = 4$ was then used to obtain position pairs from input test spectrograms.

Table 5.3 displays localization errors in the horizontal and vertical coordinates, in pixels. For WN+S mixtures, localization error for the white noise and the speech source are shown separately. As expected, performances of all the algorithms relying on WDO, namely PCESSI, VESSL and MESSL-G, are degraded using mixtures with 100% time overlap. Note that this degradation is not so strong in speech+speech mixtures using PCESSL, suggesting that this algorithm is somewhat more robust to relatively small source overlaps. This maybe explain by the fact that PCESSL has a more detailed information about the interaural manifold, since all training points are kept. It is therefore able to account for small cue perturbations due to an interferer. On the other hand, both VESSL and MESSL approximate this manifold: VESSL by using a piecewise affine approximation, MESSL by assuming a constant time delays over frequencies with only small perturbations.

Best CoL results were obtained using ILPD features. As expected, CoL performs very well on WN+WN mixtures, because it was trained with similar data. In fact, it does generally better at localizing white-noise, probably because it was used for training and provides binaural features at all TF bins. As expected, other methods performed poorly in WN+WN mixtures, since WDO is strongly violated in that case.

However, more surprisingly, CoL generally outperforms all the other methods in terms of accuracy, even when the overlap between sources is less, such as mixture of speech signals. It even yields good results in speech localization in the very challenging case of

WN+S mixtures, despite an average speech-to-noise ratio of $0 \pm 0.5\text{dB}$. The accuracy of CoL on such challenging binaural mixtures in realistic conditions, *i.e.* real world recordings in a reverberating room, is probably unequalled to date, to the best of our knowledge.

Computational times of PCESSL, VESSL ($K = 32$), MESSL, PHAT and CoL ($K^* = 100$) for a one second test mixture were respectively $0.27 \pm 0.01\text{s}$, $10.4 \pm 0.1\text{s}$, $46.7 \pm 1.2\text{s}$ and $2.2 \pm 0.1\text{s}$ using MATLAB. CoL is therefore suitable for real-time applications, while being much more accurate than all the other methods. This may not be the case for PCESSL, VESSL and MESSL, due to their costly EM iterations. While the offline training of CoL requires a computationally costly EM procedure, co-localization is straightforward and fast using the closed-form expression (4.17) of iPPAM.

We tested the influence of the number of affine components K^* and training set size N^* on CoL's performance. By Figure 5.5 (left), K^* can be tuned based on a trade-off between computation time and accuracy. Choosing $K^* = 20$ brings down the co-localization time of a 1 second mixture to 0.42 seconds, while increasing the localization error by only 6.5% relative to $K^* = 100$. Figure 5.5 (b) shows that localization error increases when N^* decreases. However, using $N^* = 5,000$ increases the mean localization error by only 3.2% relative to $N^* = 20,000$. This suggests that a less dense grid of points could be used for simpler and more practical training. While manually recording 432 positions (allowing 93,528 possible source pairs) took 22 minutes, a training set of 100 positions (allowing 5,050 source pairs) could be recorded in 5 minutes.

We finally examined the behavior of CoL in two extreme cases. First, we tested the approach on mixtures of two *equal* sound sources, *i.e.*, recordings of two loudspeakers emitting the same TIMIT utterance at the same time from two different directions. In that case, the two sources are completely overlapping, and their acoustic level ratio is constant over the entire TF plane. Over the 19 test mixtures (38 localization tasks), CoL yielded an average error of 34 pixels in the horizontal axis, 46 in the vertical axis, and 1 outlier. This is similar to results obtained on S+S mixtures with distinct speech signals (Table 5.3). On the other hand, the 3 other methods failed to localize at least one of the two sources (more than 250 pixels error) in more than half of these tests. Second, we tested the approach on 100 *non-overlapping* mixtures, *i.e.*, two consecutive 500ms speech segments emitted from different directions. Results obtained with all 4 methods were similar to those obtained for S+S mixtures in Table 5.3. Although ILD and IPD cues depend on the relative spectra of emitting sources, these last experiments suggest that CoL is quite robust to various types of time-frequency overlap in the mixtures.

5.7 Conclusion

In this chapter, we proposed three different methods for supervised multiple sound source localization. Both PCESSL and VESSL outperform state-of-the-art separation scores from MESSL and perform accurate 2D localization in the challenging case of noisy real-world recordings of multiple sparse sound sources emitting from a wide range of directions. Besides, the co-localization method yielded unexpected and surprisingly good

results. The results show that direct localization of a source pair is possible, even when the sources strongly overlap in the time-frequency plane. Contrary to prior multi-SSL methods, this is achieved without relying on the WDO assumption, and without spatially clustering binaural cues. Building a theoretical framework to better understand the mixing conditions under which co-localization performs best is an open question for future research. One may also study extensions to more sources, as well as ways of estimating the number of sources.

In general the three proposed methods push forward the concept of supervised learning as a promising way to robustly address the long-known sound source separation and localization problem using a training or *calibration* stage. The good separation results obtained with binary-masking may be viewed as a proof-of-concept suggesting that even better separation performances could be achieved. Indeed, all three methods accurately estimate the sound sources positions, which could be used to retrieve their spatial covariance matrices. This information is generally hard to get automatically and could help sound source separation algorithms that are based on Wiener filtering, *e.g.* [Févotte 05, Duong 09, Ozerov 10, Duong 10].

More generally, modeling the *physical* space where sources may emit from could greatly improve sound source separation methods. This is already suggested by Figure 5.4: While both MESSL and PCESSL rely on binary masking, PCESSL yields greater separation performance when sources share the same azimuthal plane. This is because the possible elevations of a sound source and associated interaural cue variations are explicitly taken into account in the model, while MESSL only takes into account the sources' TDOA to model interaural cues.

Further than this, the acoustic space mapping approach could be used to handle the problem of over-parameterization occurring in modern source separation methods such as [Duong 10]. This approach estimates a full-rank spatial covariance matrix for each source and at each frequency. The search-space for these matrices is of very large dimension, making parameter estimation impossible without additional knowledge on the sources. Our approach suggest that the space of spatial covariance matrices *possibly emitted by physical sources* is actually of much lower dimension. Restricting the parameters search-space to that of *physically possible* acoustic cues for a given system, *i.e.* the acoustic space, seems a worthwhile direction to improve real-world sound source separation.

Appendix

5.A Detailed Derivations of VESSL

This appendix details the derivation of closed-form expressions for VESSL’s E-XZ step (5.21), E-U step (5.22) and M-step (5.23) as well as the computation of the variational free energy to check the algorithm’s convergence.

E-XZ step: From now on, we denote more specifically by $\mathbb{E}q$ the expectation with respect to a probability distribution q . The update of $q_{X,Z}$ is given by:

$$q_{X,Z}^{(i)}(\mathbf{x}, \mathbf{z}) \propto \exp \mathbb{E}q_U^{(i-1)}[\log p(\mathbf{x}, \mathbf{z} | \mathcal{S}, \mathbf{u}; \boldsymbol{\theta}_{\text{mix}}^{(i)})]. \quad (5.26)$$

Using Bayes’ inversion and the hierarchical model decomposition (5.18) we obtain:

$$q_{X,Z}^{(i)}(\mathbf{x}, \mathbf{z}) \propto \exp \mathbb{E}q_U^{(i-1)} \left[\log \frac{p(\mathcal{S} | \mathbf{x}, \mathbf{z}, \mathbf{u}; \boldsymbol{\theta}_{\text{mix}}^{(i)}) p(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta}_{\text{mix}}^{(i)})}{p(\mathcal{S} | \mathbf{u}; \boldsymbol{\theta}_{\text{mix}}^{(i)})} \right]. \quad (5.27)$$

Using the fact that $p(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta}_{\text{mix}}^{(i)})$ does not depend on \mathbf{u} and the independence between source positions (5.15) we obtain:

$$q_{X,Z}^{(i)}(\mathbf{x}, \mathbf{z}) \propto \prod_{m=1}^M p(\mathbf{x}_m, z_m; \boldsymbol{\theta}_{\text{mix}}^{(i)}) \exp \mathbb{E}q_U^{(i-1)} \left[\log \frac{p(\mathcal{S} | \mathbf{x}, \mathbf{z}, \mathbf{u}; \boldsymbol{\theta}_{\text{mix}}^{(i)})}{p(\mathcal{S} | \mathbf{u}; \boldsymbol{\theta}_{\text{mix}}^{(i)})} \right]. \quad (5.28)$$

We can now calculate the expectation term by decomposing probabilities of \mathcal{S} into products over spectrograms observations. This can be done on the numerator because the diagonal Σ assumption (4.10) implies that spectrogram observations are independent given \mathbf{x} and \mathbf{z} , and on the denominator because observations are supposed independent and identically distributed. It follows:

$$q_{X,Z}^{(i)}(\mathbf{x}, \mathbf{z}) \propto \prod_{m=1}^M p(\mathbf{x}_m, z_m; \boldsymbol{\theta}_{\text{mix}}^{(i)}) \exp \sum_{m,d,t} q_{U_{dt}}^{(i-1)} \log \frac{p(y'_{dt} | \mathbf{x}_m, z_m, U_{dt} = m; \boldsymbol{\theta}_{\text{mix}}^{(i)})}{p(y'_{dt} | U_{dt} = m; \boldsymbol{\theta}_{\text{mix}}^{(i)})} \quad (5.29)$$

where q_U has been decomposed as $\{q_{U_{dt}}\}_{d=1,t=1}^{D,T}$ and $q_{U_{dt}}$ is defined to be 0 when $\chi_{dt} = 0$ to simplify notations. This can be re-written:

$$q_{X,Z}^{(i)}(\mathbf{x}, \mathbf{z}) \propto \prod_{m=1}^M \left(p(\mathbf{x}_m, z_m; \boldsymbol{\theta}_{\text{mix}}^{(i)}) \prod_{d=1,t=1}^{D,T} \frac{p(y'_{dt} | \mathbf{x}_m, z_m, U_{dt} = m; \boldsymbol{\theta}_{\text{mix}}^{(i)})^{q_{U_{dt}}^{(i-1)}}}{p(y'_{dt} | U_{dt} = m; \boldsymbol{\theta}_{\text{mix}}^{(i)})^{q_{U_{dt}}^{(i-1)}}} \right). \quad (5.30)$$

One may now notice that each term in the product over m is exactly the product of (4.21) and (4.25) in the details derivations of iPPAM (Appendix 4.A), except that χ_{dt} is replaced

by $q_{U_{dt}}^{(i-1)}$, the current estimated probability that source m has generated observation y'_{dt} . We deduce that

$$q_{X,Z}^{(i)}(\mathbf{x}, \mathbf{z}) = \prod_{m=1}^M q_{Z_m}^{(i)}(k) q_{X_m|Z_m}^{(i)}(\mathbf{x}_m, k) = \prod_{m=1}^M \nu_{km}^{(i)} \mathcal{N}(\mathbf{x}_m; \boldsymbol{\mu}_{km}^{(i)}, \mathbf{V}_{km}^{(i)}) \quad (5.31)$$

where

$$\boldsymbol{\mu}_{km}^{(i)} = \mathbf{V}_{km}^{(i)} \left(\tilde{\boldsymbol{\Gamma}}_k^{-1} \tilde{\mathbf{c}}_k + \sum_{d,t=1}^{D,T} \frac{q_{U_{dt}}^{(i-1)}(m)}{\sigma_d^2} (y_{dt} - \tilde{b}_{dk}) \tilde{\mathbf{a}}_{dk} \right), \quad (5.32)$$

$$\mathbf{V}_{km}^{(i)} = \left(\tilde{\boldsymbol{\Gamma}}_k^{-1} + \sum_{d,t=1}^{D,T} \frac{q_{U_{dt}}^{(i-1)}(m)}{\sigma_d^2} \tilde{\mathbf{a}}_{dk} \tilde{\mathbf{a}}_{dk}^\top \right)^{-1}, \quad (5.33)$$

$$\nu_{km}^{(i)} \propto \frac{|\mathbf{V}_{km}^{(i)}|^{\frac{1}{2}}}{|\tilde{\boldsymbol{\Gamma}}_k|^{\frac{1}{2}}} \exp -\frac{1}{2} \left(\sum_{d,t=1}^{D,T} \frac{q_{U_{dt}}^{(i-1)}(m)}{\sigma_d^2} (y_{dt} - \tilde{b}_{dk})^2 + \tilde{\mathbf{c}}_k^\top \tilde{\boldsymbol{\Gamma}}_k^{-1} \tilde{\mathbf{c}}_k - \boldsymbol{\mu}_{km}^{(i)\top} \mathbf{V}_{km}^{(i-1)} \boldsymbol{\mu}_{km}^{(i)} \right) \quad (5.34)$$

and $\nu_{km}^{(i)}$ is normalized to sum to 1 over k . One can see this step as the *localization step*, since it corresponds to estimating a mixture of Gaussians over the source position space for each source m . When $M = 1$, \mathbf{U} is entirely determined and $q_{U_{dt}} = \chi_{dt}$. We can then directly obtain the probability density $q_{X,Z}$ of the sound source position using the E-XZ step and we recover exactly iPPAM's formulas (4.12), (4.13) and (4.14) for single sound source localization. This connection between VESSL's localization step and iPPAM confirms that the proposed mPPAM model and associated variational EM procedure are natural extensions of iPPAM.

E-U step: The update of q_U is given by:

$$q_U^{(i)}(\mathbf{u}) \propto \exp \mathbb{E} q_{X,Z}^{(i)}[\log p(\mathbf{u} | \mathcal{S}, \mathbf{x}, \mathbf{z}; \boldsymbol{\theta}_{\text{mix}}^{(i)})]. \quad (5.35)$$

Using the fact that $p(\mathbf{u}; \boldsymbol{\theta}_{\text{mix}}^{(i)})$ does not depend on \mathbf{x} and \mathbf{z} and the independence between masking variables (5.16) we obtain:

$$q_U^{(i)}(\mathbf{u}) \propto \prod_{\chi_{dt}=1} p(U_{dt}; \boldsymbol{\theta}_{\text{mix}}^{(i)}) \exp \left(\mathbb{E} q_{X,Z}^{(i)}[\log p(y'_{dt} | u_{dt}, \mathbf{x}_{u_{dt}}, z_{u_{dt}}; \boldsymbol{\theta}_{\text{mix}}^{(i)})] \right) \quad (5.36)$$

For all d, t, m such that $\chi_{dt} = 1$, we can calculate $E_{dtm} = \mathbb{E} q_{X_m, Z_m}^{(i)}[\log p(y'_{dt} | U_{dt} = m, \mathbf{x}_m, z_m; \boldsymbol{\theta}_{\text{mix}}^{(i)})]$. Using the identity $\mathbb{E} q_{X_m, Z_m}^{(i)}[\cdot] = \mathbb{E} q_{Z_m}^{(i)}[\mathbb{E} q_{X_m|Z_m}^{(i)}[\cdot]]$ we obtain:

$$E_{dtm} = \sum_{k=1}^K \nu_{km}^{(i)} \mathbb{E} q_{X_m|Z_m}^{(i)}[\log p(y'_{dt} | U_{dt} = m, \mathbf{x}_m, Z_m = k; \boldsymbol{\theta}_{\text{mix}}^{(i)})] \quad (5.37)$$

$$= -\log(2\pi\sigma_d^2)/2 - \sum_{k=1}^K \frac{\nu_{km}^{(i)}}{2\sigma_d^2} \mathbb{E} q_{X_m|Z_m}^{(i)}[(y'_{dt} - \tilde{\mathbf{a}}_{dk}^\top \mathbf{x}_m - \tilde{b}_{dk})^2]. \quad (5.38)$$

According to (5.31), $q_{X_m|Z_m}^{(i)} \sim \mathcal{N}(\boldsymbol{\mu}_{km}^{(i)}, \mathbf{V}_{km}^{(i)})$. Using standard formulas for the expectation of a quadratic form of a Gaussian variable, we obtain the final E-U step update formula for $q_{U_{dt}}^{(i)}(m)$:

$$q_{U_{dt}}^{(i)}(m) \propto \frac{\lambda_{dm}^{(i)}}{\sqrt{2\pi\sigma_d^2}} \prod_{k=1}^K \exp -\frac{\nu_{km}^{(i)}}{2\sigma_d^2} \left(\text{tr}(\mathbf{V}_{km}^{(i)} \tilde{\mathbf{a}}_{dk} \tilde{\mathbf{a}}_{dk}^\top) + (y_{dt} - \tilde{\mathbf{a}}_{dk}^\top \boldsymbol{\mu}_{km}^{(i)} - \tilde{b}_{dk})^2 \right) \quad (5.39)$$

where each $q_{U_{dt}}^{(i)}$ is normalized to sum to 1 over m . Note that $q_{U_{dt}}$ is defined for $\chi_{dt} = 1$ only, and $q_{U_{dt}}$ is set to 0 when $\chi_{dt} = 0$ for convenience, as explained in the E-XZ step. The E-U step can be seen as the *sound source separation step*, as it provides the masking variable probabilities, and hence allow to deduce a binary mask for each source.

M step: We need to maximize the expected complete-data log-likelihood:

$$\boldsymbol{\theta}_{\text{mix}}^{(i+1)} = \underset{\boldsymbol{\theta}_{\text{mix}}}{\text{argmax}} \mathbb{E}_{q_U^{(i)} q_{X,Z}^{(i)}} [\log p(\mathcal{S}, \mathbf{u}, \mathbf{x}, \mathbf{z}; \boldsymbol{\theta}_{\text{mix}})]. \quad (5.40)$$

This reduces to the update of noise variances $\boldsymbol{\Sigma}^{(i)} = \text{diag}(\sigma_1^{2(i)} \dots \sigma_D^{2(i)})$ and sources' weights $\boldsymbol{\lambda}^{(i)}$. By finding zeros of (5.40)'s derivatives, we find:

$$\lambda_{dm}^{(i)} = \frac{1}{\bar{\chi}_{dt}} \sum_{t=1}^T q_{U_{dt}}^{(i)}(m) \quad \text{where } \bar{\chi}_d = \sum_{t=1}^T \chi_{dt} \quad \text{and} \quad (5.41)$$

$$\sigma_d^{2(i)} = \frac{\sum_{t,m,k} q_{U_{dt}}^{(i)}(m) \nu_{km}^{(i)} \left(\text{tr}(\mathbf{V}_{km}^{(i)} \tilde{\mathbf{a}}_{dk} \tilde{\mathbf{a}}_{dk}^\top) + (y_{dt} - \tilde{\mathbf{a}}_{dk}^\top \boldsymbol{\mu}_{km}^{(i)} - \tilde{b}_{dk})^2 \right)}{\sum_{t,m,k} q_{U_{dt}}^{(i)}(m) \nu_{km}^{(i)}}. \quad (5.42)$$

Convergence check: In a variational EM context, the quantity to monitor at each step and to maximize is the variational free energy \mathcal{E} defined by

$$\mathcal{E}(q^{(i)}, \boldsymbol{\theta}_{\text{mix}}^{(i)}) = \mathbb{E}_q \left[\log \frac{p(\mathcal{S}, \mathbf{x}, \mathbf{z}, \mathbf{u}; \boldsymbol{\theta}_{\text{mix}})}{q(\mathbf{x}, \mathbf{z}, \mathbf{u})} \right]. \quad (5.43)$$

Recall that $q^{(i)}(\mathbf{u}, \mathbf{x}, \mathbf{z})$ denotes the estimated missing variable posterior distribution, and is iteratively optimized using the variational approximation (5.20). By construction, \mathcal{E} increases at each iteration and is a lower bound of the observed-data log-likelihood. We consider that the algorithm converged at iteration i , and hence reached a local maximum of the log-likelihood, when \mathcal{E} increased by less than 0.1% of its total increase since the beginning of the algorithm at iteration i . The variational energy at each iteration decomposes as:

$$\begin{aligned} \mathcal{E}(q, \boldsymbol{\theta}_{\text{mix}}) &= \sum_{\chi_{dt}=1} \mathbb{E}_{q_{U_{dt}}} [\log p(y'_{dt} | \mathbf{x}, \mathbf{z}, u_{dt}; \boldsymbol{\theta}_{\text{mix}})] + \sum_{m=1}^M \mathbb{E}_{q_{X_m, Z_m}} \left[\log \frac{p(\mathbf{x}_m, z_m; \theta)}{q(\mathbf{x}_m, z_m)} \right] \\ &+ \sum_{\chi_{dt}=1} \mathbb{E}_{q_{U_{dt}}} \left[\log \frac{p(U_{dt}; \theta)}{q(U_{dt})} \right] \end{aligned} \quad (5.44)$$

where the iteration superscripts ⁽ⁱ⁾ have been omitted to simplify notations. We see that $\mathcal{E}(q, \boldsymbol{\theta}_{\text{mix}})$ decomposes into a sum of three terms $S_1 + S_2 + S_3$ that can be calculated successively. The first term is:

$$S_1 = \sum_{\chi_{dt}=1} \sum_{m=1}^M q_{U_{dt}}(m) \log \frac{\tilde{q}_{U_{dt}}(m)}{\lambda_{dm}} \quad (5.45)$$

where $\tilde{q}_{U_{dt}}(m)$ denotes the E-U step update *before normalization*, as given in (5.39). The second term is:

$$S_2 = - \sum_{k=1, m=1}^{K, M} \nu_{km} \log \nu_{km} + KL[\mathcal{N}(\mathbf{x}_m; \boldsymbol{\mu}_{km}, \mathbf{V}_{km}); \mathcal{N}(\mathbf{x}_m; \tilde{\mathbf{c}}_k, \tilde{\boldsymbol{\Gamma}}_k)] \quad (5.46)$$

where the term $KL[\dots]$ is the Kullback-Leibler divergence between two multivariate normal distributions and is equal to

$$\frac{1}{2} \left(\text{tr}(\tilde{\boldsymbol{\Gamma}}_k^{-1} \mathbf{V}_{km}) + (\tilde{\mathbf{c}}_k - \boldsymbol{\mu}_{km})^\top \tilde{\boldsymbol{\Gamma}}_k^{-1} (\tilde{\mathbf{c}}_k - \boldsymbol{\mu}_{km}) - \log \frac{|\mathbf{V}_{km}|}{|\tilde{\boldsymbol{\Gamma}}_k|} - L \right). \quad (5.47)$$

Finally, the third term is:

$$S_3 = - \sum_{\chi_{dt}=1} \sum_{m=1}^M q_{U_{dt}}(m) \log \frac{q_{U_{dt}}(m)}{\lambda_{dm}}. \quad (5.48)$$

Note that $S_1 + S_3$ simplifies as:

$$S_1 + S_3 = - \sum_{\chi_{dt}=1} \log \left(\sum_{m=1}^M \tilde{q}_{U_{dt}}(m) \right). \quad (5.49)$$

6.1 Summary and Discussion

We addressed the long-studied problem of binaural sound source separation and localization through an original approach, based on learning. To achieve so, we first defined the central concept of acoustic space, and developed a new methodological framework to gather large acoustic space datasets. These datasets could be used to prove a fundamental property of acoustic space: they are smooth, locally-linear, low-dimensional manifolds parameterized by sound source directions.

With this key property in mind we presented a general family of probabilistic models for locally-linear high- to low-dimensional mappings, referred to as *Gaussian locally-linear mapping* (GLLiM). Several insights on GLLiMs were provided, including a connection to joint Gaussian mixture models and a discussion on forward versus inverse mapping strategies. We justified the advantage of inverse mapping in high- to low-dimensional regression problems, and subsequently developed a particular instance of GLLiM referred to as *probabilistic piecewise affine mapping* (PPAM). A more general model referred to as partially-latent-output mapping (PLOM) was also proposed. PLOM was showed to unify a number of existing regression and dimensionality reduction methods, while generalizing them to situations where some of the output's components cannot be observed. The prominent advantage of both PPAM and PLOM methods was demonstrated on a large number of tasks beyond the scope of auditory scene analysis, including synthetic function inversion, face pose and light estimation from images and retrieval of physical properties from Mars hyperspectral data.

We then addressed the problem of sound source localization (SSL) based on training datasets. Two methods were proposed. The first one, called probabilistic piecewise-constant mapping (PPCM), maybe be viewed as a probabilistic extension of nearest-neighbor, allowing to deal with noisy time series of vector input with missing values. The second one, called inverse PPAM, consists in an extension of PPAM based on Baye's rule. Both showed an unequaled accuracy in two-dimensional binaural SSL on real-world data.

Finally, we tackled the more challenging problem of multiple SSL in binaural sound mixtures. Three approaches were proposed. The first two approaches rely on the W-disjoint orthogonality assumption, *i.e.*, only one source is dominating at each time-frequency point. They may be viewed as mixed version of the PPCM and PPAM models. These extensions yielded closed form expectation-maximization procedures alternating between binary-masking separation and source localization. The third approach, called co-localization, radically contrasts with existing multiple SSL methods, and showed unexpectedly good results in source-pair localization. Co-localization does not rely on WDO, and use the iPPAM method to directly map a mixed interaural spectrogram to a 4-dimensional vector containing the 2D positions of both sources. The three methods showed to dramatically outperform state-of-the-art binaural separation and localization techniques on real-world data.

In summary, this work pushes forward acoustic space mapping as a promising framework in computational auditory scene analysis. Inspired by the observation that humans learned auditory perception through experience, we showed that learning could have a strong impact in computational audition and presented a new range of models, protocols and techniques that bridge the gap between some machine learning tools and traditional binaural signal processing methods. An intrinsic novelty of this work is to model the physical position of sound sources through random variables in spaces corresponding to other modalities, namely motor states or pixel-coordinates in a camera. This open the way to new connections between the intensively studied field of space mapping in machine learning on the one hand, and the intensively studied field of computational auditory scene analysis on the other hand.

A limitation of proposed approaches is that they require a training or calibration phase. Hence, they do not perform completely *blind* source separation or localization, in contrast with many existing approaches. For this reason, proposed methods are limited to specific application, in which a training stage is actually possible. For instance, this is not the case when the task is to post-process music or movies soundtrack. But this limitation may as well be viewed as an advantage when the goal is to take audio signal processing methods *out of the lab*, to real world scenarios. The proposed framework differs from many other methods in that it is *intrinsically designed* to deal with real world data. Most existing sound source separation or localization techniques initially rely on theoretical models of the sound propagation and mixing process. Their ability to deal with recordings made in a real-world environment is only validated *a posteriori*. In fact, quite often, they are not evaluated on real world data but rather on virtual acoustic environments that can be simulated by software such as Roomsim [Campbell 05]. In this thesis, a somewhat contrary approach was employed. The ground bases of this work were obtained from recordings in a real room *i.e.*, the acoustic space of our binaural system. Rather than depending on the ability of an initial model to approximate the real world, the performance of proposed methods depend on how much recording conditions varied between the training and the testing stage. This may both be viewed as an advantage or as a limitation, depending on the specific task addressed. Scenarios in which acoustic space mapping is believed to have a great practical interest are those occurring in a real-world place, when prior calibration

is possible. This could include live music recording on a stage or concert hall, speech localization, diarisation or enhancement in a meeting or conference room, or hearing aid devices (the system could be calibrated for a specific wearer).

6.2 Direction for Future Research

Rather than an end, we would like to view this thesis as a starting point for fascinating future research topics. We propose here a non-exhaustive list of possible follow-ups.

- An important direction is to study more thoroughly the influence of changes in experimental conditions on binaural manifolds. What happens when changing the position of the recording setup? Moving to another room? What is the influence of the sound source distance and directivity? What happens when the HRTFs change? While the PLOM model is a possible direction to improve robustness to such situations, other methods such as *transfer learning* [Pan 10] could be envisioned. A more ambitious idea would be to learn acoustic spaces in virtual environments, using a room simulator such as Roomsim [Campbell 05]. One could imagine learning many different models in different room configurations, *e.g.*, microphones position, room size, reverberations. When dealing with real world data, the most appropriate model could be selected from virtually learned one using, *e.g.*, model selection techniques.
- In our view, the surprisingly good results obtained with the co-localization method open the doors to a new category of binaural processing methods, and deserve a deeper understanding. First of all, a new theoretical framework need to be built to understand why the algorithm performs so well. Then, many possible extensions could be envision: Should we include the ratio of power between the two sources during learning? Can the approach be extended to mixture of three sources? One could consider learning acoustic spaces for mixture of sources from all possible directions, with various acoustic level at each position. This could be viewed as a model for diffuse sounds. Alternatively, some training set of diffuse sounds could be built directly from real world data, *e.g.*, by placing the binaural system in a crowded environment. More generally, co-localization results suggests that the spatial richness of binaural cues has not yet been fully exploited, and might allow to deal with much more complex auditory scenes. The key question to ask is then how to select the best model in order to automatically determine the number and type of sound sources?
- The problem of localizing and tracking moving sound sources is of great practical interest, but few approaches exist due to its difficulty. We believe that the probabilistic models underlying our methods constitute an adequate tool to handle such situations. For example, one may consider adding some hidden Markov model on the source position variables over time in the mixed PPAM model and the VESSL algorithm.

-
- Some parameters of the proposed models must be tuned manually. Most notably the number of sources M , the number of affine transformation K and the number of latent components L_w . Automatically estimating them based on probabilistic criteria is a challenging yet worthwhile direction for future research.
 - While all the models, data and experiments in this thesis were designed for a binaural system, nothing intrinsically restricts them to two microphones. The most straightforward way to extend them to more microphones would be to concatenate interaural vectors from the different microphone pairs, but more sophisticated ways could be envisioned.
 - The low computational time and robustness to real world conditions showed by iP-PAM in both 1 and 2-source localization allow to envision real-time implementations for interactive systems. New questions would then emerged, such as how to automatically fit the analysis window on perceived signals, or how to automatically estimate whether a source emits within or outside the trained area.
 - Instead of recording a static white noise source at different positions, one could imagine spanning the space continuously with the emitter. Preliminary results of section 4.4.1 suggest that PPAM can deal with single spectrogram window rather than temporal means in the training stage. However, the number of training data would be much bigger at the window level. Numerically, this would require to scale up the PPAM training procedure to work with bigger datasets in reasonable time.
 - In this thesis, we separately used audio-motor and audio-visual learning procedures for auditory scene analysis. A fascinating question is how to connect these procedures together? Achieving this could yield a unified probabilistic framework for audio-visuo-motor perception and interaction. One could then imagine *closing the sensorimotor loop* by allowing the system to perform voluntary motor actions based on perceived auditory and visual signals.
-

PUBLICATIONS

International Conference Publications

- [Deleforge 12b] Antoine Deleforge & Radu Horaud. *The Cocktail Party Robot: Sound Source Separation and Localisation with an Active Binaural Head*. In IEEE/ACM International Conference on Human Robot Interaction, pages 431–438, Boston, Massachusetts, March 2012.
- [Deleforge 12c] Antoine Deleforge & Radu Horaud. *A Latently Constrained Mixture Model for Audio Source Separation and Localization*. In proceedings of the 10th International Conference on Latent Variable Analysis and Signal Separation, volume LNCS 7191, pages 372–379, Tel Aviv, Israel, March 2012.
- [Deleforge 12a] Antoine Deleforge & Radu Horaud. *2D Sound-Source Localization on the Binaural Manifold*. In IEEE International Workshop on Machine Learning for Signal Processing, pages 1–6, Santander, Spain, September 2012.
- [Deleforge 13b] Antoine Deleforge, Florence Forbes & Radu Horaud. *Variational EM for Binaural Sound-Source Separation and Localization*. In IEEE International Conference on Acoustic, Speech, Signal Processing, Vancouver, Canada, May 2013.

International Journal Submissions

- Antoine Deleforge, Florence Forbes & Radu Horaud. *Hearing on Binaural Manifolds: Acoustic Space Learning for Sound-Source Separation and Localization*. International Journal of Neural Systems. Submitted in May 2013.
- Antoine Deleforge, Yoav Y. Schechner, Laurent Girin & Radu Horaud. *Binaural Co-Localization of Audio Source Pairs*. IEEE Signal Processing Letters. Submitted in July 2013.
- Antoine Deleforge, Florence Forbes & Radu Horaud. *Mapping Learning with Partially-Latent Output*. Statistics and Computing. Springer. Submitted in July 2013.

Other Articles

- [Deleforge 11] Antoine Deleforge & Radu Horaud. *Learning the Direction of a Sound Source Using Head Motions and Spectral Features*. Research Report RR-7529, INRIA, February 2011.
 - [Sanchez-Riera 12] Jordi Sanchez-Riera, Xavier Alameda-Pineda, Johannes Wienke, Antoine Deleforge, Soraya Arias, Jan Cech, Sebastian Wrede & Radu Horaud P. *Online Multimodal Speaker Detection for Humanoid Robots*. In IEEE International Conference on Humanoid Robotics (Humanoids), Osaka, Japan, December 2012.
 - [Alameda-Pineda 13] Xavier Alameda-Pineda, Jordi Sanchez-Riera, Johannes Wienke, Vojtech Franc, Jan Cech, Kaustubh Kulkarni, Antoine Deleforge & Radu Horaud. *RAVEL: An Annotated Corpus for Training Robots with Audiovisual Abilities*. Journal on Multimodal User Interfaces, vol. 7, no. 1-2, pages 79–91, 2013.
 - [Deleforge 13a] Antoine Deleforge, Florence Forbes & Radu Horaud. *Mapping Learning with Partially Latent Output*. arXiv preprint arXiv:1308.2302, August 2013.
 - [Cech 13a] Jan Cech, Ravi-Kant Mittal, Antoine Deleforge, Jordi Sanchez-Riera, Xavier Alameda-Pineda & Radu Horaud. *Active-Speaker Detection and Localization with Microphones and Cameras Embedded into a Robotic Head*. In IEEE International Conference on Humanoid Robots, Atlanta, United States, September 2013.
-

REFERENCES

- [Aarabi 02] P. Aarabi. *Self-localizing dynamic microphone arrays*. IEEE Trans. Syst., Man, Cybern. C, vol. 32, no. 4, pages 474–484, 2002.
- [Adragni 09] K. P. Adragni & R. D. Cook. *Sufficient dimension reduction and prediction in regression*. Philosophical Transactions of the Royal Society A, vol. 367, no. 1906, pages 4385–4405, 2009.
- [Agarwal 04] A. Agarwal & B. Triggs. *Learning to track 3D human motion from silhouettes*. In C. E. Brodley, editeur, 21st International Conference on Machine Learning (ICML '04), volume 69, pages 9–16, Banff, Canada, 2004. ACM Press.
- [Agarwal 06] A. Agarwal & B. Triggs. *Recovering 3D human pose from monocular images*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 1, pages 44–58, 2006.
- [Alameda-Pineda 12] X. Alameda-Pineda & R. P. Horaud. *Geometrically Constrained Robust Time Delay Estimation Using Non-coplanar Microphone Arrays*. In Proceeding of the 20th European Signal Processing Conference (EUSIPCO), pages 1309–1313, 2012.
- [Alameda-Pineda 13] X. Alameda-Pineda, J. Sanchez-Riera, J. Wienke, V. Franc, J. Cech, K. Kulkarni, A. Deleforge & R. Horaud P. *RAVEL: An Annotated Corpus for Training Robots with Audiovisual Abilities*. Journal on Multimodal User Interfaces, vol. 7, no. 1-2, pages 79–91, 2013.
- [Avnimelech 99] R. Avnimelech & N. Intrator. *Boosted mixture of experts: an ensemble learning scheme*. Neural computation, vol. 11, no. 2, pages 483–497, 1999.
- [Aytekin 08] M. Aytekin, C. F. Moss & J. Z. Simon. *A Sensorimotor Approach to Sound Localization*. Neural Computation, vol. 20, no. 3, pages 603–635, 2008.

- [Bach 05] F. R. Bach & M. I. Jordan. *A probabilistic interpretation of canonical correlation analysis*. Rapport technique 688, Department of Statistics, University of California, Berkeley, 2005.
- [Beal 03] M. Beal & Z. Ghahramani. *The variational Bayesian EM Algorithm for incomplete data: with application to scoring graphical model structures*. Bayesian Statistics, pages 453–464, 2003.
- [Belkin 03] M. Belkin & P. Niyogi. *Laplacian eigenmaps for dimensionality reduction and data representation*. Neural computation, vol. 15, no. 6, pages 1373–1396, 2003.
- [Bensaid 10] S. Bensaid, A. Schutz & D. T. M. Slock. *Single microphone blind audio source separation using EM-Kalman filter and short+long term AR modeling*. In Latent Variable Analysis and Signal Separation, pages 106–113, 2010.
- [Bernard-Michel 09] C. Bernard-Michel, S. Douté, M. Fauvel, L. Gardes & S. Girard. *Retrieval of Mars surface physical properties from OMEGA hyperspectral images using Regularized Sliced Inverse Regression*. Journal of Geophysical Research: Planets, vol. 114, no. E6, page 6005, 2009.
- [Bibring 04] J.-P. Bibring, A. Soufflot & B. M. *OMEGA: Observatoire pour la Minéralogie, l’Eau, les Glaces et l’Activité, in Mars Express: The Scientific Payload*. vol. ESA SP-1240, page 3749, 2004.
- [Bishop 98] C. M. Bishop, M. Svensén & C. K. I. Williams. *GTM: The generative topographic mapping*. Neural computation, vol. 10, no. 1, pages 215–234, 1998.
- [Blauert 97] J. Blauert. *Spatial hearing: The psychophysics of human sound localization*. MIT Press, 1997.
- [Bofill 03] P. Bofill. *Underdetermined blind separation of delayed sound sources in the frequency domain*. Neurocomputing, vol. 55, no. 3, pages 627–641, 2003.
- [Bouveyron 11] C. Bouveyron, G. Celeux & S. Girard. *Intrinsic dimension estimation by maximum likelihood in isotropic probabilistic PCA*. Pattern Recognition Letters, vol. 32, pages 1706–1713, 2011.
- [Buchner 05] H. Buchner, R. Aichner & W. Kellermann. *A generalization of blind source separation algorithms for convolutive mixtures based on second-order statistics*. IEEE TASLP, vol. 13, no. 1, pages 120–134, 2005.
- [Calvert 04] G. A. Calvert, C. Spence & B. E. Stein. *The handbook of multi-sensory processes*. MIT press, 2004.

-
- [Campbell 05] D. Campbell, K. Palomaki & G. Brown. *A Matlab simulation of "shoebox" room acoustics for use in research and teaching*. Computing and Information Systems, vol. 9, no. 3, page 48, 2005.
- [Cardoso 93] J.-F. Cardoso & A. Soudoukiac. *Blind beamforming for non-Gaussian signals*. In IEE Proceedings F (Radar and Signal Processing), volume 140, pages 362–370. IET, 1993.
- [Cech 13a] J. Cech, R. Mittal, A. Deleforge, J. Sanchez-Riera, X. Alameda-Pineda & R. Horaud P. *Active-Speaker Detection and Localization with Microphones and Cameras Embedded into a Robotic Head*. In IEEE International Conference on Humanoid Robots, Atlanta, United States, September 2013. IEEE Robotics Society.
- [Cech 13b] J. Cech, R. Mittal, A. Deleforge, J. Sanchez-Riera, X. Alameda-Pineda & R. P. Horaud. *Active-Speaker Detection and Localization with Microphones and Cameras Embedded into a Robotic Head*. In IEEE International Conference on Humanoid Robots, Atlanta, USA, October 2013. IEEE Robotics and Automation Society.
- [Celeux 92] G. Celeux & G. Govaert. *A classification EM algorithm for clustering and two stochastic versions*. Comp. Stat. & Data An., vol. 14, no. 3, pages 315–332, 1992.
- [Cherry 53] E. C. Cherry. *Some experiment on the recognition of speech, with one and with two ears*. JASA, vol. 25, no. 5, pages 975–979, September 1953.
- [Comon 10] P. Comon & C. Jutten. *Handbook of Blind Source Separation, Independent Component Analysis and Applications*. Academic Press (Elsevier), 2010.
- [Cook 07] R. D. Cook. *Fisher Lecture: Dimension Reduction in Regression*. Statistical Science, vol. 22, no. 1, pages 1–26, 2007.
- [de Veaux 89] R. D. de Veaux. *Mixtures of linear regressions*. Comput. Stat. Data Anal., vol. 8, no. 3, pages 227–245, 1989.
- [Deleforge 11] A. Deleforge & R. Horaud. *Learning the Direction of a Sound Source Using Head Motions and Spectral Features*. Research Report RR-7529, INRIA, February 2011.
- [Deleforge 12a] A. Deleforge & R. P. Horaud. *2D Sound-Source Localization on the Binaural Manifold*. In IEEE Int. Workshop Machine Learning for Signal Processing, pages 1–6, Santander, Spain, September 2012.
- [Deleforge 12b] A. Deleforge & R. P. Horaud. *The Cocktail Party Robot: Sound Source Separation and Localisation with an Active Binaural Head*. In IEEE/ACM International Conference on Human Robot Interaction, pages 431–438, Boston, Mass, March 2012.

- [Deleforge 12c] A. Deleforge & R. P. Horaud. *A Latently Constrained Mixture Model for Audio Source Separation and Localization*. In Proceedings of the 10th International Conference on Latent Variable Analysis and Signal Separation, volume LNCS 7191, pages 372–379, Tel Aviv, Israel, March 2012. Springer-Verlag.
- [Deleforge 13a] A. Deleforge, F. Forbes & R. Horaud. *Mapping Learning with Partially Latent Output*. arXiv preprint arXiv:1308.2302, August 2013.
- [Deleforge 13b] A. Deleforge, F. Forbes & R. P. Horaud. *Variational EM for Binaural Sound-Source Separation and Localization*. In IEEE Int. Conf. Acoust., Speech, Signal Process., Vancouver, Canada, May 2013.
- [Douté 05] S. Douté, B. Schmitt, J.-P. Bibring, Y. Langevin, F. Altieri, G. Bellucci, B. Gondet & the MEX OMEGA team. *Nature and composition of the icy terrains of the south pole of Mars from MEX OMEGA observations*. In 36th Lunar and Planetary Science Conference, (Lunar and Planetary Science XXXVI), page 1734, March 2005.
- [Douté 07] S. Douté, E. Deforas, F. Schmidt, R. Oliva & B. Schmitt. *A Comprehensive Numerical Package for the Modeling of Mars Hyperspectral Images*. In 38th Lunar and Planetary Science Conference, (Lunar and Planetary Science XXXVIII), page 1836, League City, Texas, March 2007.
- [Duong 09] N. Q. Duong, E. Vincent & R. Gribonval. *Spatial covariance models for under-determined reverberant audio source separation*. In Applications of Signal Processing to Audio and Acoustics, 2009. WASPAA'09. IEEE Workshop on, pages 129–132. IEEE, 2009.
- [Duong 10] N. Duong, E. Vincent & R. Gribonval. *Under-determined reverberant audio source separation using a full-rank spatial covariance model*. IEEE TASLP, vol. 18, no. 7, pages 1830–1840, 2010.
- [Févotte 05] C. Févotte & J.-F. Cardoso. *Maximum likelihood approach for blind audio source separation using time-frequency Gaussian source models*. In Applications of Signal Processing to Audio and Acoustics, 2005. IEEE Workshop on, pages 78–81. IEEE, 2005.
- [Fusi 12] N. Fusi, O. Stegle & N. D. Lawrence. *Joint modelling of confounding factors and prominent genetic regulators provides increased accuracy in genetical genomics studies*. PLoS computational biology, vol. 8, no. 1, page e1002330, 2012.
- [Garofolo 93] J. S. Garofolo, L. F. Lamel & W. M. Fisher. *The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus CDRM*, 1993.

-
- [Ghahramani 96] Z. Ghahramani & G. E. Hinton. *The EM algorithm for mixtures of factor analyzers*. Rapport technique CRG-TR-96-1, University of Toronto, 1996.
- [Ghazanfar 06] A. A. Ghazanfar & C. E. Schroeder. *Is neocortex essentially multisensory?* Trends in cognitive sciences, vol. 10, no. 6, pages 278–285, 2006.
- [Giacinto 00] G. Giacinto & F. Roli. *Dynamic classifier selection*. In Multiple Classifier Systems, pages 177–189. Springer, 2000.
- [Güler 05] I. Güler & E. D. Übeyli. *A mixture of experts network structure for modelling Doppler ultrasound blood flow signals*. Computers in Biology and Medicine, vol. 35, no. 7, pages 565–582, 2005.
- [Harding 06] S. Harding, J. Barker & G. J. Brown. *Mask estimation for missing data speech recognition based on statistics of binaural interaction*. Audio, Speech, and Language Processing, IEEE Transactions on, vol. 14, no. 1, pages 58–67, 2006.
- [Haykin 05] S. Haykin & Z. Chen. *The Cocktail Party Problem*. Neural Computation, vol. 17, pages 1875–1902, 2005.
- [Held 63] R. Held & A. Hein. *Movement-produced stimulation in the development of visually guided behavior*. J. Comp. Physiol. Psych., vol. 56, no. 5, pages 872–876, 1963.
- [Hofman 98a] P. M. Hofman & A. J. Van Opstal. *Spectro-temporal factors in two-dimensional human sound localization*. JASA, vol. 103, no. 5, pages 2634–2648, 1998.
- [Hofman 98b] P. M. Hofman, J. G. Van Riswick & A. J. Van Opstal. *Relearning sound localization with new ears*. Nature neuroscience, vol. 1, no. 5, pages 417–421, 1998.
- [Hörnstein 06] J. Hörnstein, M. Lopes, J. Santos-victor & F. Lacerda. *Sound localization for humanoid robots building audio-motor maps based on the HRTF. CONTACT project report*. In Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pages 1170–1176, 2006.
- [Huerta 03] G. Huerta, W. Jiang & M. A. Tanner. *Time series modeling via hierarchical mixtures*. Statistica Sinica, vol. 13, no. 4, pages 1097–1118, 2003.
- [Jordan 94] M. Jordan & R. Jacobs. *Hierarchical mixtures of experts and the EM algorithm*. Neural computation, vol. 6, no. 2, pages 181–214, 1994.

- [Kain 98] A. Kain & M. Macon. *Spectral voice conversion for text-to-speech synthesis*. In proc. ICASSP, volume 1, 1998.
- [Kalaitzis 12] A. Kalaitzis & N. Lawrence. *Residual component analysis: Generalising PCA for more flexible inference in linear-Gaussian models*. In ICML, Edinburgh, Scotland, UK, 2012.
- [Keyrouz 07] F. Keyrouz, W. Maier & K. Diepold. *Robotic Localization and Separation of Concurrent Sound Sources Using Self-Splitting Competitive Learning*. In Proc. of IEEE CIISP, pages 340–345, 2007.
- [Khan 13] M. S. Khan, S. M. Naqvi, A. ur Rehman, W. Wang & J. Chambers. *Video-aided model-based source separation in real reverberant rooms*. IEEE Transactions on Audio, Speech, and Language Processing, vol. 21, no. 9, pages 1900–1912, September 2013.
- [Kidron 05] E. Kidron, Y. Y. Schechner & M. Elad. *Pixels that sound*. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, pages 88–95. IEEE, 2005.
- [Kullaib 09] A. R. Kullaib, M. Al-Mualla & D. Vernon. *2D Binaural Sound Localization: for Urban Search and Rescue Robotics*. In proc. Mobile Robotics, pages 423–435, 2009.
- [Lawrence 05] N. Lawrence. *Probabilistic non-linear principal component analysis with Gaussian process latent variable models*. The Journal of Machine Learning Research, vol. 6, pages 1783–1816, 2005.
- [Lee 10] S.-Y. Lee & H.-M. Park. *Multiple Reverberant Sound Localization Based on Rigorous Zero-Crossing-Based ITD Selection*. IEEE Signal Process. Lett., vol. 17, no. 7, pages 671–674, 2010.
- [Li 91] K. C. Li. *Sliced Inverse Regression for Dimension Reduction*. Journal of the American Statistical Association, vol. 86, no. 414, pages 316–327, 1991.
- [Liu 10] R. Liu & Y. Wang. *Azimuthal source localization using interaural coherence in a robotic dog: modeling and application*. Robotica, vol. 28, no. 7, pages 1013–1020, 2010.
- [Mandel 07] M. I. Mandel, D. P. W. Ellis & T. Jebara. *An EM Algorithm for Localizing Multiple Sound Sources in Reverberant Environments*. In Proc. NIPS, pages 953–960, 2007.
- [Mandel 10] M. I. Mandel, R. J. Weiss & D. P. W. Ellis. *Model-based expectation-maximization source separation and localization*. IEEE Trans. Acoust., Speech, Signal Process., vol. 18, no. 2, pages 382–394, 2010.

-
- [Markovich 09] S. Markovich, S. Gannot & I. Cohen. *Multichannel eigenspace beamforming in a reverberant noisy environment with multiple interfering speech signals*. IEEE Transactions on Audio, Speech, and Language Processing, vol. 17, no. 6, pages 1071–1086, 2009.
- [Middlebrooks 91] J. C. Middlebrooks & D. M. Green. *Sound Localization by Human Listeners*. Annual Review of Psychology, vol. 42, pages 135–159, 1991.
- [Mouba 06] J. Mouba & S. Marchand. *A source localization/separation/respatialization system based on unsupervised classification of interaural cues*. In Proceedings of the International Conference on Digital Audio Effects, pages 233–238, 2006.
- [Naik 00] P. Naik & C.-L. Tsai. *Partial least squares estimator for single-index models*. Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 62, no. 4, pages 763–771, 2000.
- [O’Regan 01] J. K. O’Regan & A. Noe. *A sensorimotor account of vision and visual consciousness*. Behavioral and Brain Sciences, vol. 24, pages 939–1031, 2001.
- [Otani 09] M. Otani, T. Hirahara & S. Ise. *Numerical study on source-distance dependency of head-related transfer functions*. JASA, vol. 125, no. 5, pages 3253–61, 2009.
- [Ozerov 10] A. Ozerov & C. Févotte. *Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation*. Audio, Speech, and Language Processing, IEEE Transactions on, vol. 18, no. 3, pages 550–563, 2010.
- [Pan 10] S. J. Pan & Q. Yang. *A Survey on Transfer Learning*. IEEE Transactions on Knowledge and Data Engineering, vol. 22, pages 1345–1359, 2010.
- [Parra 02] L. C. Parra & C. V. Alvino. *Geometric source separation: Merging convolutive source separation with geometric beamforming*. Speech and Audio Processing, IEEE Transactions on, vol. 10, no. 6, pages 352–362, 2002.
- [Peng 96] F. Peng, R. A. Jacobs & M. A. Tanner. *Bayesian inference in mixtures-of-experts and hierarchical mixtures-of-experts models with an application to speech recognition*. Journal of the American Statistical Association, vol. 91, no. 435, pages 953–960, 1996.
- [Poincaré 29] H. Poincaré. *The foundations of science; science and hypothesis, the value of science, science and method*. New York: Science Press, 1929. Halsted, G. B. trans. of *La valeur de la science*, 1905.

- [Qiao 09] Y. Qiao & N. Minematsu. *Mixture of Probabilistic Linear Regressions: A unified view of GMM-based mapping techniques*. In proc. ICASSP, pages 3913–3916, 2009.
- [Radfar 07] M. H. Radfar & R. M. Dansereau. *Single-channel speech separation using soft mask filtering*. Audio, Speech, and Language Processing, IEEE Transactions on, vol. 15, no. 8, pages 2299–2310, 2007.
- [Rayleigh 07] L. Rayleigh. *On our perception of sound direction*. Philos. Mag., vol. 13, pages 214–232, April 1907.
- [Roman 03] N. Roman, D. Wang & G. J. Brown. *Speech segregation based on sound localization*. J. Acoust. Soc. Am., vol. 114, no. 4, pages 2236–2252, 2003.
- [Rosipal 06] R. Rosipal & N. Krämer. *Overview and recent advances in partial least squares*. In C. Saunders, M. Grobelnik, S. Gunn & J. Shawe-Taylor, editors, Subspace, Latent Structure and Feature Selection, volume 3940 of *Lecture Notes in Computer Science*, pages 34–51. Springer Berlin Heidelberg, 2006.
- [Roweis 00] S. T. Roweis. *One Microphone Source Separation*. In Advances in Neural Information Processing Systems, volume 13, pages 793–799. MIT Press, 2000.
- [Sanchez-Riera 12] J. Sanchez-Riera, X. Alameda-Pineda, J. Wienke, A. Deleforge, S. Arias, J. Cech, S. Wrede & R. Horaud P. *Online Multimodal Speaker Detection for Humanoid Robots*. In IEEE International Conference on Humanoid Robotics (Humanoids), Osaka, Japan, December 2012.
- [Saul 03] L. Saul & S. Roweis. *Think globally, fit locally: unsupervised learning of low dimensional manifolds*. Journal of Machine Learning Research, vol. 4, pages 119–155, 2003.
- [Sawada 07] H. Sawada, S. Araki & S. Makino. *A Two-Stage Frequency-Domain Blind Source Separation Method for Underdetermined Convolutional Mixtures*. In Proc. of WASPAA, 2007.
- [Schmidt 06] M. Schmidt & R. Olsson. *Single-channel speech separation using sparse non-negative matrix factorization*. 2006.
- [Scholkopf 98] B. Scholkopf, A. J. Smola & K. R. Muller. *Nonlinear Component Analysis as a Kernel Eigenvalue Problem*. Neural Computation, vol. 10, pages 1299–1319, 1998.

-
- [Senkowski 08] D. Senkowski, T. R. Schneider, J. J. Foxe & A. K. Engel. *Cross-modal binding through neural coherence: implications for multi-sensory processing*. Trends in neurosciences, vol. 31, no. 8, pages 401–409, 2008.
- [Smaragdis 97] P. Smaragdis. *Efficient blind separation of convolved sound mixtures*. In Applications of Signal Processing to Audio and Acoustics, 1997. 1997 IEEE ASSP Workshop on, pages 4–pp. IEEE, 1997.
- [Smaragdis 09] P. Smaragdis, M. Shashanka & B. Raj. *A sparse non-parametric approach for single channel separation of known sounds*. In Advances in Neural Information Processing Systems, pages 1705–1713, 2009.
- [Smola 04] A. J. Smola & B. Schölkopf. *A tutorial on support vector regression*. Statistics and computing, vol. 14, no. 3, pages 199–222, 2004.
- [Talmon 11] R. Talmon, I. Cohen & S. Gannot. *Supervised source localization using diffusion kernels*. In Workshop on Applications of Signal Processing to Audio and Acoustics, pages 245–248, 2011.
- [Tenenbaum 00] J. B. Tenenbaum, V. de Silva & J. C. Langford. *A global geometric framework for nonlinear dimensionality reduction*. Science, vol. 290, pages 2319–2323, 2000.
- [Thayananthan 06] A. Thayananthan, R. Navaratnam, B. Stenger, P. Torr & R. Cipolla. *Multivariate relevance vector machines for tracking*. In European Conference on Computer Vision, pages 124–138. Springer, 2006.
- [Tipping 99a] M. E. Tipping & C. M. Bishop. *Mixtures of probabilistic principal component analyzers*. Neural Computation, vol. 11, no. 2, pages 443–482, February 1999.
- [Tipping 99b] M. E. Tipping & C. M. Bishop. *Probabilistic principal component analysis*. Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 61, no. 3, pages 611–622, 1999.
- [Tipping 01] M. Tipping. *Sparse Bayesian learning and the relevance vector machine*. The Journal of Machine Learning Research, vol. 1, pages 211–244, 2001.
- [Vapnik 97] V. Vapnik, S. Golowich & A. Smola. *Support vector method for function approximation, regression estimation, and signal processing*. In Advances in Neural Information Processing Systems 9 — Proceedings of the 1996 Neural Information Processing Systems Conference (NIPS 1996), pages 281–287. MIT Press, Cambridge, MA, USA, December 1997.

- [Vincent 06] E. Vincent, R. Gribonval & C. Févotte. *Performance measurement in blind audio source separation*. IEEE TASLP, vol. 14, no. 4, pages 1462–1469, 2006.
- [Viste 03] H. Viste & G. Evangelista. *On the Use of Spatial Cues to Improve Binaural Source Separation*. In proc. DAFX, pages 209–213, 2003.
- [Wahba 90] G. Wahba. Spline models for observational data. Numeéro 59. Siam, 1990.
- [Wang 05] W. Wang, D. Cosker, Y. Hicks, S. Saneit & J. Chambers. *Video assisted speech source separation*. In Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on, volume 5, pages v–425. IEEE, 2005.
- [Wang 06] D. Wang & G. J. Brown. Computational auditory scene analysis: Principles, algorithms and applications. IEEE Press, 2006.
- [Wang 12] C. Wang & R. M. Neal. *Gaussian Process Regression with Heteroscedastic or Non-Gaussian Residuals*. Computing Research Repository, vol. abs/1212.6246, 2012.
- [Winter 07] S. Winter, W. Kellermann, H. Sawada & S. Makino. *MAP-based underdetermined blind source separation of convolutive mixtures by hierarchical clustering and L1-norm minimization*. EURASIP Journal on Advances in Signal Processing, vol. 2007, pages 1–12, January 2007. Article ID 24717.
- [Woodruff 12] J. Woodruff & D. Wang. *Binaural localization of multiple sources in reverberant and noisy environments*. IEEE Trans. Acoust., Speech, Signal Process., vol. 20, no. 5, pages 1503–1512, 2012.
- [Woodworth 65] R. S. Woodworth & H. Schlosberg. Experimental psychology. Holt, 1965.
- [Wright 06] B. A. Wright & Y. Zhang. *A review of learning with normal and altered sound-localization cues in human adults*. International journal of audiology, vol. 45, no. S1, pages 92–98, 2006.
- [Wu 08] H. Wu. *Kernel sliced inverse regression with applications to classification*. Journal of Computational and Graphical Statistics, vol. 17, no. 3, pages 590–610, 2008.
- [Xu 95] L. Xu, M. I. Jordan & G. E. Hinton. *An Alternative Model for Mixtures of Experts*. In proc. NIPS, volume 7, pages 633–640, 1995.
- [Yilmaz 04] O. Yilmaz & S. Rickard. *Blind separation of speech mixtures via time-frequency masking*. IEEE Transactions on Signal Processing, vol. 52, pages 1830–1847, 2004.

- [Zhang 04] Z. Zhang & H. Zha. *Principal manifolds and nonlinear dimensionality reduction via tangent space alignment*. SIAM Journal on Scientific Computing, vol. 26, no. 1, 2004.
- [Zhigljavsky 08] A. Zhigljavsky & A. Žilinskas. *Stochastic global optimization*. Springer, 2008.