FAST NUMERICAL AND MACHINE LEARNING ALGORITHMS FOR SPATIAL AUDIO REPRODUCTION

by

Yuancheng Luo

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2014

Advisory Committee:
Professor Ramani Duraiswami, Chair/Advisor
Dr. Dmitry N. Zotkin
Professor Larry Davis
Professor Hal Daumé III
Professor David Jacobs
Professor Shihab Shamma, Dean's Representative

ABSTRACT

Title of dissertation: FAST NUMERICAL AND
MACHINE LEARNING ALGORITHMS
FOR SPATIAL AUDIO REPRODUCTION

Yuancheng Luo, Doctor of Philosophy, 2014

Dissertation directed by: Professor Ramani Duraiswami
Department of Computer Science

Audio reproduction technologies have underwent several revolutions from a purely mechanical, to electromagnetic, and into a digital process. These changes have resulted in steady improvements in the objective qualities of sound capture/playback on increasingly portable devices. However, most mobile playback devices remove important spatial-directional components of externalized sound which are natural to the subjective experience of human hearing. Fortunately, the missing spatial-directional parts can be integrated back into audio through a combination of computational methods and physical knowledge of how sound scatters off of the listener's anthropometry in the sound-field. The former employs signal processing techniques for rendering the sound-field. The latter employs approximations of the sound-field through the measurement of so-called Head-Related Impulse Responses/Transfer Functions (HRIRs/HRTFs).

This dissertation develops several numerical and machine learning algorithms for accelerating and personalizing spatial audio reproduction in light of available mobile computing power. First, spatial audio synthesis between a sound-source and sound-field

requires fast convolution algorithms between the audio-stream and the HRIRs. We introduce a novel sparse decomposition algorithm for HRIRs based on non-negative matrix factorization that allows for faster time-domain convolution than frequency-domain fast-Fourier-transform variants. Second, the full sound-field over the spherical coordinate domain must be efficiently approximated from a finite collection of HRTFs. We develop a joint spatial-frequency covariance model for Gaussian process regression (GPR) and sparse-GPR methods that supports the fast interpolation and data fusion of HRTFs across multiple data-sets. Third, the direct measurement of HRTFs requires specialized equipment that is unsuited for widespread acquisition. We "bootstrap" the human ability to localize sound in listening tests with Gaussian process active-learning techniques over graphical user interfaces that allows the listener to infer his/her own HRTFs. Experiments are conducted on publicly available HRTF datasets and human listeners.

Acknowledgments

The dissertation is the product of the accumulated efforts, intuitions, and influences from my many interactions with literature, my advisor, and fellow researchers. First, I would like to acknowledge the massive body of scientific works between the fields of spatial audio, numerical methods, and machine learning for providing the foundations of which this dissertation stands upon. This includes the numerous pioneers in those fields, their contributions, and the easy accessibility to their works through portals such as Google scholar, Wikipedia, and the University of Maryland network.

Second, I would like to acknowledge my advisor, mentor, and friend, Professor Ramani Duraiswami for all the ideas, first drafts, and philosophical ramblings that we've exchanged. Without his flexible demeanor, optimistic attitude, and intellectual feedback, I would be walking on a whole different path in another field. Third, I would like to thank my fellow researchers and colleagues for making this a positive learning experience: Dr. Zotkin for finding errors and translating much of my later drafts into professional pieces of writing. Dr. Gumerov for teaching me about fast multipole methods. Qi Hu for getting me into physical fitness and providing job hunting tips. Balaji Srinivasan for supporting and accommodating me to my first conference. Ross Adelman for listening to my job hunting rants. Last, I'd like to thank the remaining committee members, Dr. Larry Davis, Dr. Hal Daumé III, Dr. David Jacobs, and Dr. Shihab Shamma for taking the time to read this work. Of course, this dissertation would have not been possible without the following funding sources: National Science Foundation (Award IIS-1117716), Office of Naval Research (MURI grant N00014-08-10638).

Table of Contents

List of Figures vii

List of Abbreviations xi

1 Introduction 1
- 1.1 Spatial Audio Reproduction and Applications . . . 4
- 1.2 Head-Related Transfer Function (HRTF) . . . 7
  - 1.2.1 Min-phase Representation . . . 8
  - 1.2.2 Measurement Methods and Costs . . . 9
- 1.3 Sound-Fields and Spherical Interpolants . . . 15
- 1.4 Spatial Audio Synthesis and Playback . . . 18
- 1.5 Machine Learning . . . 20
- 1.6 Organization . . . 21

2 Gaussian Process Models for Sound-Source Localization and Active-Learning 24
- 2.1 Introduction . . . 24
  - 2.1.1 Prior Works . . . 25
  - 2.1.2 Present Work . . . 26
- 2.2 Formulation of Problems . . . 28
  - 2.2.1 Feature subset-selection . . . 29
  - 2.2.2 Active-learning for individualizing HRTFs . . . 30
- 2.3 Binaural Sound-Source Invariant Features . . . 32
- 2.4 Gaussian Process Regression for SSL . . . 34
  - 2.4.1 Choice of Covariance Functions . . . 37
  - 2.4.2 Model-Order and Cost Analysis . . . 38
  - 2.4.3 Experiments . . . 40
- 2.5 Feature Subset-Selection . . . 43
  - 2.5.1 Incremental GP Models . . . 44
  - 2.5.2 GP L^2 Risk Function Criterions . . . 45
  - 2.5.3 Experiments . . . 47
- 2.6 Active-Learner System . . . 48
  - 2.6.1 Conditional Mixture of Gaussians Models . . . 49

2.6.2 GPs for Modeling SSLE . . . 51
2.6.3 Query-Selection . . . 52
2.6.4 Experiments . . . 53
2.7 Conclusions . . . 56
2.8 Appendix: Matérn Product Integrals . . . 57

3 Fast Sparse and Gridded Gaussian Process Regression for HRTF Interpolation 58
3.1 Introduction . . . 58
3.2 GPR Background . . . 62
3.2.1 Kronecker Product Methods for GPR . . . 65
3.2.2 Grid GPR and Cost Analysis . . . 67
3.2.3 Relation to GPLVM . . . 69
3.3 Sparse-Grid GPR . . . 70
3.4 Missing Data for Grid GPR . . . 74
3.5 Extra Data for Grid GPR . . . 78
3.6 Fast Greedy Backward Subset Selection . . . 80
3.7 Experiments and Applications . . . 84
3.7.1 Performance Tests on Synthetic Data . . . 85
3.7.2 Grid and Sparse-Grid GPs for HRTF Interpolation . . . 88
3.7.2.1 Grid and Sparse-Grid GPs Comparisons . . . 90
3.7.2.2 Cross-Validation Experiments . . . 92
3.7.2.3 Kernel Function Series Expansions . . . 94
3.7.2.4 Spectral-Extrema Extraction . . . 97
3.7.3 Greedy Backward Subset Selection for Time-Delay Supports . . . 100
3.7.4 Greedy Backward Subset Selection for HRTFs . . . 102
3.8 Conclusions . . . 105
3.9 Appendix . . . 106
3.9.1 Kronecker Product Identities . . . 106
3.9.2 Relation to GPLVM . . . 107
3.9.3 Economical DTC . . . 107
3.9.4 Missing Data DTC . . . 109
3.9.5 Index Operations . . . 109
3.9.6 Spherical Covariance Function Representations . . . 110

4 Heterogeneous HRTF Dataset Fusion via Gaussian Processes 111
4.1 Introduction . . . 111
4.1.1 Problem Formulation . . . 114
4.2 Gaussian Process Regression . . . 115
4.2.1 Spatial-Frequency Covariance Functions for Sound-Fields . . . 117
4.3 Data Fusion and Transformations . . . 119
4.3.1 Equalization-Transform . . . 120
4.3.2 Window-Transform . . . 122
4.3.3 Composition . . . 123
4.4 Experiments . . . 123
4.5 Conclusions . . . 126

4.6 Acknowledgment . . . 127

5 Efficient Multicore Non-negative Least Squares 128
5.1 Introduction . . . 128
5.1.1 Non-negative Least Squares . . . 130
5.1.2 Survey of NNLS Algorithms . . . 131
5.2 Active-set Method . . . 132
5.3 Proposed Algorithm . . . 134
5.3.1 QR Updating by Modified Gram-Schmidt . . . 135
5.3.2 Alternative QR Updating by Rotations . . . 137
5.3.3 Alternative QR Updating by Semi-normal Equations . . . 138
5.3.4 QR Downdating by Rotations . . . 139
5.4 Multi-core CPU Architectures . . . 141
5.4.1 CPU Implementation . . . 143
5.5 GPU Architectures . . . 143
5.5.1 GPU Implementation . . . 146
5.5.2 Parallelizing QR Methods . . . 147
5.5.3 Memory Usage . . . 148
5.6 Applications for Deconvolution and Time-Delay Estimation . . . 150
5.7 Experiments . . . 152
5.7.1 Synthetic Deconvolution . . . 154
5.7.2 Non-Synthetic Deconvolution . . . 155
5.7.3 Interaural Time Difference Estimation . . . 156
5.8 Conclusions . . . 160

6 Sparse Head-Related Impulse Response for Efficient Direct Convolution 162
6.1 Introduction . . . 162
6.2 Problem Formulation . . . 164
6.3 Semi-non-negative Toeplitz Matrix Factorization . . . 166
6.3.1 Background . . . 166
6.3.2 Notational Conventions . . . 168
6.3.3 Toeplitz-Constrained Semi-NMF . . . 169
6.3.4 Minimizing the Number of Reflections . . . 171
6.4 Experiments and Results . . . 174
6.4.1 HRIR/HRTF Data Information . . . 174
6.4.2 Error Metric . . . 175
6.4.3 Resonance and Reflection Filter Training . . . 176
6.4.4 Regularization Term Influence . . . 177
6.4.5 Transformation Bandwidth Optimization . . . 179
6.4.6 Computational Cost . . . 181
6.5 Discussion . . . 181
6.6 Conclusions . . . 183

7 Conclusions 184
7.1 Open Problems 185
7.1.1 Toeplitz Matrix Factorizations for Blind-Dereverberation 185
7.1.2 Alternative Covariance Functions for Gaussian Processes 187
7.1.3 Perceptual Measures of HRTF Similarity 188

Bibliography 190

List of Figures

1.1 Spherical coordinate system and the direct-measurement process with mics in left and right ears (left). Pinna anthropometry features (right, image courtesy of [1]). 8
1.2 Log-magnitude HRTFs are smooth along spatial and frequency domains (horizontal and vertical plane directions are shown). 10
1.3 Playback along (θ, ϕ) via interpolated min-phase HRTFs with time-delay. 19

2.1 Gaussian Process Regression with binaural features (bottom two boxes) to perform two types of inferences. On the left are shown the steps needed to perform sound-source localization. On the right is shown an active-learning framework that combines SSL with listening tests to learn a listener's HRTFs. 29
2.2 Binaural features extracted from CIPIC subject 156 HRTFs are shown for horizontal and median plane directions. 34
2.3 Cumulative energy of leading eigenvalues for K are shown for GP-SSL models (varying covariance functions and feature types). 41
2.4 Mercator projections of GP-SSL K_∞ predicted mean directions evidenced on randomized and subset-selected inputs (prediction error risk function R in section 2.5.2 are shown. 42
2.5 Generalization errors are shown for GP-SSL models evidenced on randomized (dotted) and GFS [prediction error (solid), normalized error (dashed)] selected subsets of feature-direction pairs. 48
2.6 GUI shows a mercator projection of spherical coordinate system onto 2D panel. User clicks on panel to report a direction. 49
2.7 Distribution (box-plot) of hyperparameter values are shown for GP-SSL models (x-axis $0-22.1$ kHz frequency range). Large valued hyperparameters ℓ_k indicate less sensitivity along the k^{th} frequency. 52
2.8 Nearest localized directions after active-learning by the GP-SSL model (red) improve upon initial non-individualized HRTF localizations (blue). 55
2.9 Mean angular errors are shown for the initial query (non-individualized HRTFs) and nearest HRTF queries. 55

3.1 Posterior distributions for standard grid and sparse-DTC grid GP methods are shown for synthetic data (Eq. 3.34). Initial sparse inputs (□) once trained (X) move away from the origin in the 2D case. 86
3.2 Training runtimes, LMH, and RMSE are shown for cases of varying dimension D (fixed $m = 32$, $M = 32^D$) and varying number of inputs per dimension 2^m (fixed dimension $D = 2$, $M = (2^m)^2$) for standard, DTC, standard grid, and sparse-DTC grid GP methods. 87
3.3 Training runtimes, LMH, and RMSE are shown for cases of varying number of missing data R and number of extra data S for fixed dimension $D = 2$ and fixed number of inputs (32^2) across standard, DTC, and standard grid GP methods. 88
3.4 Posterior mean magnitude responses and variances for grid and sparse-DTC-grid GPR along the azimuth plane are shown. Initial inducing inputs are marked as □ and trained inputs are marked as **X**. 91
3.5 Learning curves (runtime, LMH, and SD) for grid and sparse-DTC-grid GP methods for are shown for increasing number of inducing inputs. Lower SD indicates more accurate predictions. 93
3.6 Spectral distortion (SD) errors are shown for predictions made by grid and sparse-DTC-grid GPs across 45 CIPIC subject right-ear HRTF datasets. Sparse cases consist of 100 and 50 inducing inputs (optimized in spherical domain, fixed in frequency). 94
3.7 Cumulative SDRs (dB) for the interpolation (random partition) and extrapolation (missing hole) experiments are shown for grid GPR, inverse distance, spherical harmonic, and spline interpolants. Larger SDRs indicate more accurate predictions. 95
3.8 CIPIC measurement directions are mapped to a spherical coordinate grid under a simple rotation. 97
3.9 Approximation errors are shown for the series expansion of the squared exponential of chordal distance for both varying number of truncation terms ρ and varying hyperparameters ℓ. 98
3.10 GP and sparse-GP posterior distributions for the magnitude HRTF responses are shown. Spectral extrema are extracted from the zero-crossing of the posterior mean's gradient. 99
3.11 Kernel density estimation (Gaussian) of pooled spectral extrema for GPs trained on horizontal and median plane HRTFs across all subjects, right ears. 100
3.12 GP predicted ITD means are shown in the left-column; GP predicted ITD variances (95% confidence) are shown in the right-column. The measurement directions are marked ∘. Predictions evidenced on the full and GBSS ITD measurements belong to the top and bottom-rows respectively. 101
3.13 Learning curves are shown for RMSE prediction errors (left plot) and remaining data LMH (right) for GPs evidenced on the GBSS remaining samples as inputs are removed. 102

3.14 Grid GP's posterior magnitude response means are shown for inputs (azimuth plane and $8.8 - 13$ kHz inputs). The models are evidenced on the remaining subset-selected inputs (eliminated inputs are marked **X**). "Salient/redundant" refer to selection-strategies that "minimize/maximize" the remaining data LMH respectively; plots labeled "removal" and "reconstruction" fill in the missing inputs via grid GP inference. Bottom-row plots show the trade-off between subset-sizes and SD (over all predictions). Low SD indicates small error. 103
3.15 Average grid GP's (specified on the azimuth plane at different frequency ranges) remaining data LMH and SD (over all predictions) are shown for increasing subset-selected sizes. Low SD indicates small error. 104

4.1 Mercator projection of measurement grids are shown for "Club Fritz" Neumann HRTFs. For anonymity, the source institutions are indicated by the lab numbers. 112
4.2 Reference GPs are trained and whose covariance function is reused for a second GP specified over the combined datasets. The transformation filter coefficients are trained and the fused sound-field is given by the second GP conditioned on the transformed datasets. 120
4.3 Plots in row 1 are the reference HRTFs (labs $1 - 7$) on horizontal plane (x-axis $-\pi < \phi < \pi$ and y-axis $0 < \omega < 18$ kHz). Plots along different columns refer to the reference dataset in row 1. Rows 2 and 3 are the sound-fields (GP predicted magnitude response means conditioned on non-transformed datasets and transformed datasets respectively). 124
4.4 Plots in row 1 are the reference HRTFs (labs $1, 2, 3, 5, 6, 7$) on median plane (x-axis $-\pi < \theta < \pi$ and y-axis $0 < \omega < 18$ kHz). Plots along different columns refer to the reference dataset in row 1. Rows 2 and 3 are the sound-fields (GP predicted magnitude response means conditioned on non-transformed datasets and transformed datasets respectively). 125
4.5 Rows 1 and 2 show the horizontal-plane-trained window-filter coefficients (after min-phase reconstruction into time-domain) and the equalization-transform coefficients (absolute log-space) respectively; Row 3 show the SDRs (w.r.t. column i reference datasets) of various sound-fields specified on different datasets: reference +, the non-transformed *, and transformed x. 126
4.6 Rows 1 and 2 show the median-plane-trained window-filter coefficients (after min-phase reconstruction into time-domain) and the equalization-transform coefficients (absolute log-space) respectively; Row 3 show the SDRs (w.r.t. column i reference datasets) of various sound-fields specified on different datasets: reference +, the non-transformed *, and transformed x. 127

5.1 GPU multiprocessor utilizes hierarchical memory model spanning fast on-chip and shared memory accessible by the local multiprocessor to slow off-ship global memory accessible by all multiprocessors. 144

5.2 ITD of left and right ear CIPIC HRIRs on the Azimuth plane for subject 3 are shown. The x-axis represents integer time-bins. 158
5.3 Horizontal plane ITD errors, computed over various methods (max-peak, cross-correlation, least squares, and NNLS), w.r.t. the Woodworth model [2] (ITD $= a(\phi + \sin\phi)/c$ for the sphere radius a anthropometric parameter $X_2/2$, sound speed c, and azimuth ϕ in radians) are shown. 159
5.4 Cross-correlation and NNLS solutions for HRIR pairs on the azimuth plane (negative, zero, and positive time-delays centered at 100 time-samples) are shown. 161

6.1 Modified semi-non-negative matrix factorization generalizes time-domain convolution for a collection of HRTFs X, resonance filter f, and non-negative reflection filters in G. 166
6.2 RMSE / SD error progress over 25 algorithm iterations. 170
6.3 Top row: Slices of reflection filter matrix G trained without sparsity constraint; also, original HRIR after min-phase processing, time delay removing, and normalization. Bottow row: Slices of reflection filter matrix G trained with sparsity constraint applied ($\lambda = 10^{-3}$); also, HRTR reconstructed from it. 172
6.4 Influence of the L_1 regularization term λ in Eq 6.14 on NNZE and on the reconstruction error for sample HRIR. 177
6.5 A map of NNZE and SD error over the full spherical coordinate range for left-ear HRIR data. Note smaller NNZE / SD values on ipsilateral side. . . 178
6.6 A comparison between varying-sparsity L_1-NNLS and L_1-LS solutions for selected directions on horizontal and median planes. Angles are listed in radians. 179
6.7 SD error dependence on bandwidth of window transform for a sample HRIR. 180

List of Abbreviations

BLAS	Basic Linear Algebra Subprograms
CUDA	Compute Unified Device Architecture
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
GBSS	Greedy Backward Subset-Selection
GFS	Greedy Forward Selection
GP	Gaussian Process
GPLVM	Gaussian Process Latent Variable Model
GPR	Gaussian Process Regression
GPU	Graphics Processing Unit
HRIR	Head-Related Impulse Response
HRTF	Head-Related Transfer Function
IID	Interaural Intensity Difference
ITD	Interaural Time Difference
KDE	Kernel Density Estimation
KTP	Kronecker Tensor Product
KTVP	Kronecker Tensor Vector Product
LMH	Log Marginal-likelihood
MGS	Modified Gram-Schmidt
MIMD	Multiple-Instruction Multiple-Data
MKL	Math Kernel Library
MoG	Mixture of Gaussians
NNLS	Non-Negative Least Squares
OpenMP	Open Message Passing
OU	Ornstein-Uhlenbeck
PCA	Principal Component Analysis
RMSE	Root Mean Squared Error
SD	Spectral Distortion
SDR	Signal-to-Distortion Ratio
SIMD	Single-Instruction Multiple-Data
SSL	Sound-Source Localization
SSLE	Sound-Source Localization Error
SVR	Support Vector Regression
TPK	Tensor Product Kernel
VAD	Virtual Auditory Display
VKTP	Vector Kronecker Tensor Product

C_h	Chordal Distance
$\circ$	Hadamard Product
$\otimes$	Kronecker Product
$\times$	Cartesian Outer-product
diag $(\cdot)$	Diagonalization
bdg $(\cdot)$	Block Diagonalization
Tps $(\cdot)$	Topelitz
tr $(\cdot)$	Trace
vec$(\cdot)$	Vectorization

Chapter 1: Introduction

Sound recording and audio playback technologies have undergone several revolutions since their inceptions over a century ago. Advances in sound recording processes have steadily improved from low to high precision and accuracy: Analog devices such as the phonograph cylinder physically transcribed changes in the sound-air pressure into grooves on metal plates during the early industrialized era. The processes soon became electronic via the transformation of the physical mechanics into electric currents and their storage on magnetic tape during the modern era; the polarity of the current can be edited and stored on the magnetic tape without significant loss in quality. The latest revolution in the digital domain allowed sound to be stored in high resolution digital representations (bits) after the invention of the microprocessor. Advances in audio playback processes have underwent similar changes in terms of fidelity and portability: Early mechanical gramophones were soon replaced by electromagnet based dynamic loudspeakers and later by increasingly portable variants such as the head/ear phone. Stereo and surround-sound reproduction standards all require a pair of head-phones or multiple loudspeakers.

While these developments have improved various technologies for objectively recording and reproducing audio content, many important subjective aspects of audio perception were ignored. One such aspect is the spatial perception/dimension of audio where sounds,

heard by humans, naturally contain cues for the direction and distances of their acoustic-sources [3]. Directional content is valuable as it parameterizes acoustic events in space much in the same way that objects can be arranged in a visual system; both have a physical origin and a direction relative to a field of view that is knowable by the subject. Moreover, this human ability to localize sound-source directions may be more valuable than other perceptual qualities such as speech intelligibility [4] and psychophysical masking [5, 6] in varying contexts; sound-source localization extends the human sensory awareness of events to regions outside the visual field and behind occlusions. For example, man's spatial auditory perception has saved him from a host of potentially lethal circumstances from the sound of an incoming automobile in one's blind-spot, to the noise of a river when water is scarce, to the footsteps of an ambushing saber-tooth tiger during the night!

Study of such phenomena belongs to the field of spatial audio where its primary concerns are directed towards the understanding of the listener's phenomenology via subjective assessments of hearing. However, an instrumental treatment of such studies would evaluate their *usefulness* for external adjustments, namely the reproduction of audio that can be spatialized by the subject through the means of technology. Unfortunately, traditional sound-recording technologies only capture the spatial characteristics of sound as recorded by receivers that are independent of the human listener; many subjective cues such as the acoustic sound paths in the near-field centered about the listener's head are not measured.

This consequent loss of spatial information is especially apparent when listening to audio over mobile devices such as head/ear phones; 3D sound is no longer externalized without recreating how sound scatters off of the listener's anthropometry such as

his/her head, torso, and outer-ears (pinnae). Recreating this sound-field is a difficult task as parts of the listener's anthropometry interfere with the sound spectrum [3]: The human head blocks high frequencies. The shape of the pinnae causes reflections and resonances that alter the spectrum of the sound along different angles of incidences. Reflections off the body/torso alter the low-frequency range. These factors are all individualized due to variations in the shape of anthropometry across the human population but can be summarized by so-called "Head-related Transfer Functions" (HRTFs) [7] which can be empirically measured. While directly measuring HRTFs requires specialized equipment that few individuals/labs possess, those with knowledge of one's HRTFs can place arbitrary sound-sources or virtual loudspeakers in 3D with great accuracy.

Fortunately, recent advances in both audio streaming rates and available processing power have created opportunities for introducing spatial information back into audio. Moreover, there's a growing demand for personalized technology that has generated new incentives for the re-integration and prediction of the user's information with regards to biometrics and preferences [8]. Such a movement is possible through wide-spread means of acquiring data through the increased connectivity between general-purpose computing devices and aggregative services. Data acquisition technologies such as cameras, headphones, and microphones are commonly built into mobile computing devices such as laptop, smart-phone, and tablets. The latter provides a medium for the recorded contents to be processed and then personalized through the organization, ranking, and selection of content by both machine and user before transferred and aggregated over a network for analysis. Subsequent knowledge discovery from the aggregated data pool is delivered back to the mobile platform through the network and their exposure to the subject

provides new feedback.

This new-found availability of data and the accessibility to mobile computing power by users has elevated two interdisciplinary fields of studies in relation to digital media. The first is the development of efficient numerical and computational methods which are relevant for real-time streaming applications on low-power mobile computing devices. The second is the development of machine learning methods for inference and personalization which are relevant for predicting the subject's preferences. For spatial audio reproduction, we show that these two disciplines provide the means for accelerating spatial audio rendering and for learning individualized sound-source localization cues and HRTFs. We explore and develop several algorithms in depth for these purposes within this dissertation.

1.1 Spatial Audio Reproduction and Applications

Spatial audio reproduction is the synthesis of sounds as if heard from a virtual source in 3D space. The field of spatial audio has a long history that begins with psychoacoustic models of how humans localized sound. One such model in literature, Lord Rayleigh's duplex theory [9], posits that the sound-source direction can be estimated by a set of binaural cues derived from sound heard by the two ears; the interaural time difference (ITD) and the interaural intensity difference (IID) define the differences in the time-of-arrival and amplitude between a sound's wavefront reaching the subject's left and right ears respectively [7]. Later models relate sound-source direction to ITD via simplified models of a spherical or ellipsoidal head [2, 10]. While these binaural cues are important

for spatialization, especially for sound-source directions that are coplanar and horizontal to the two ears (horizontal plane), they are subsumed by more complex spectral cues that result from sound scattering and reflecting off of the subject's anthropometry. The anthropometry features, most notably those that characterize the parts of the pinnae, vary considerably with the individual and thus motivate the need for personalized spatial audio [7, 11].

One method for quantifying the audio spectral cues for the individual is to physically measure *impulse responses* between the ear canals and sound-source locations in 3D often positioned over a spherical grid; an impulse response is the output (response) of a dynamic system when presented with a brief input signal (impulse). Microphones placed within an individual's ears record the impulse emitted from loudspeakers distributed over a spherical array. The recordings form the direction-dependent responses that are summarized by the so called "Head-Related Impulse Response/Transfer Function" (HRIR/HRTF). Moreover, the effects of the recording environment are removed by subtracting away a recording done in the absence of the subject; the HRTFs represent only the spectral distortions from sound scattering off the listener in an otherwise free-field.

Knowledge of an individual's HRTFs allow one to position sound-sources arbitrarily around a subject's ears for playback through headphones; several applications are possible from these assumptions. First, virtual acoustic scenes consisting of sound propagating through complex environments can be simulated in software and then heard by the subject. Such simulations generally model the sound-reflection paths off of an underlying geometric representation of the environment and then computing the time-delays and angles of reflections that would enter a virtual listener's ear. This setup may be em-

ployed in virtual acoustic displays (VADs) [12, 13] in combination with a 3D graphics engine where a polygonal decomposition of the environment and its material properties are hard-coded thus remain relatively constant during run-time. Second, realistic acoustic scenes recorded by 3D audio camera / spherical microphone array can be navigated by a subject using a head-tracker and graphical user interface (GUI) [14]. If the orientation of the listener's ear or virtual ears is known, then it is possible to filter the acoustic streams (for directional microphones) or the beam-formed acoustic streams (for omni-directional microphones) with the head-aligned HRTFs before mixing and rendering.

The general process for spatial audio reproduction can be divided into three stages. We present their respective background and literature review in the subsequent sub-sections:

1. Section 1.2 describes the limits of the conventional (direct) HRTF measurement process and alternative methods for inferring HRTF.

2. Section 1.3 describes the interpolation of a finite collection of HRTFs over the spherical coordinate domain.

3. Section 1.4 describes the synthesis/filtering of spatial audio between HRTFs and arbitrary sound-sources.

This dissertation addresses various problems that arise from each of the aforementioned stages; many of our contributions draw inspiration from the fields of machine learning, numerical linear algebra, and signal processing which we have already published in [15–23]. The organization of the remaining dissertation is presented in section 1.6.

1.2 Head-Related Transfer Function (HRTF)

Formally, an HRTF is the far-field frequency response of a listener's left or right ear measured from a specific point in the free-field in 3D to a specific point in the ear canal [24]. The effects of the measurement environment are removed by dividing out a recording in the absence of the listener in the frequency domain. HRTFs have insofar been physically measured from real subjects [1, 25, 26], computationally simulated from human meshes [27–29], inferred from human anthropometry [30–32], and tuned through listening tests [20, 33, 34].

As physical measurements, HRIRs and HRTFs are typically parameterized along a spherical coordinate system defined by azimuth ϕ (left and right) and elevation θ (up and down) directions. In the direct-measurement process, two small microphones are partially inserted into a subject's blocked ear canals and record a known broad-band spectral stimulus played by loudspeakers along a spherical grid. The loudspeakers are located along a fixed radius from the center of the head and can be specified in terms of azimuth ϕ and elevation θ angles as shown in Figure. 1.1. The radius or the loudspeaker distance from the subject is ignored as the sound-wave can be approximated by a plane-wave at larger measurement distances; the impulse from a common loudspeaker tends toward a plane-wave whose relative direction to both left and right ears are the same.

The HRIR and HRTF representations are derived as follows: Microphone recordings are outputs or the left and right ear finite impulse responses (FIRs) of a linear time-

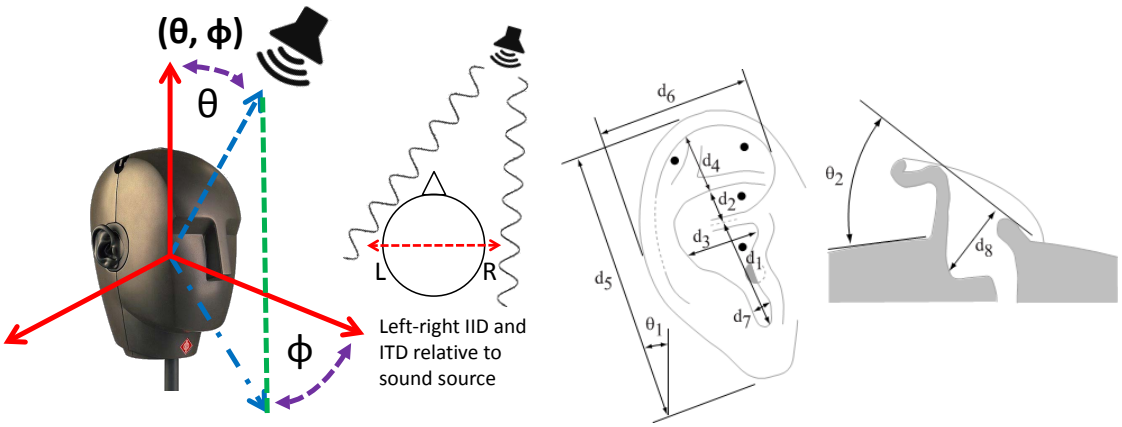


Figure 1.1: Spherical coordinate system and the direct-measurement process with mics in left and right ears (left). Pinna anthropometry features (right, image courtesy of [1]).

variant (LTI) system; the frequency-domain HRTF $H(f)$ is given by

$$H(f) = \text{Output}(f)/\text{Input}(f), \tag{1.1}$$

where f is frequency, output is the Fourier transform of the recording, and input is the Fourier transform of the free-field recording of the loudspeaker's impulse in the absence of the subject. Moreover, the HRTF $H(f)$ is the Fourier transform of the time-domain HRIR $h(t)$; the HRIR can be recovered by taking the inverse Fourier transform of the HRTF.

1.2.1 Min-phase Representation

HRTFs can be specified by their normalized minimum-phase FIR representations as the ITD and IID information can be added back into the phase and magnitude components respectively. The minimum-phase representation relates the HRTF's magnitude with its phase response via the Hilbert transform [35]; the initial time-delay is decoupled from the

FIR due to minimum group-delay and minimum energy-delay properties. In the discrete case, min-phase representation is computed from the discrete Hilbert transform of the natural log-magnitude of the HRTF given by

$$z(n) = \begin{cases} 1, & n = 0, \quad n = L/2 \\ 2, & n = 1, \ldots, L/2 - 1 \\ 0, & n = L/2 + 1, \ldots, L - 1 \end{cases}, \qquad \begin{aligned} c_{\theta,\phi} &= [\mathcal{F}^{-1}\{\log|\mathcal{F}\{h_{\theta,\phi}\}|\}] \circ z, \\ \hat{h}_{\theta,\phi} &= \mathcal{F}^{-1}\{\exp\{c_{\theta,\phi}\}\}, \end{aligned} \tag{1.2}$$

where L is the filter length of HRIR $h_{\theta,\phi}(n)$ and $\hat{h}_{\theta,\phi}(n)$ is the minimum-phase representation. This allows VADs systems to characterize auditory cues of subject-direction via the normalized left and right ear magnitude responses, ITD, and IID [36] (see section 1.4 for details). Only the first half the magnitude frequency responses are required due to symmetry reflected in the second half. The magnitude HRTFs are also smooth along both spatial and frequency domains as shown in Fig. 1.2; this property is useful for reducing the sampling density of the spherical measurement grid and model-order complexity for inference problems.

1.2.2 Measurement Methods and Costs

While HRTFs are formally acquired via direct measurements, the process has a number of drawbacks that renders dependent technologies such as VADs and applications inaccessible to a wider public. First, direct measurements require specialized equipment such as wireless microphones that fit into the ear canal and a loudspeaker array that is typically

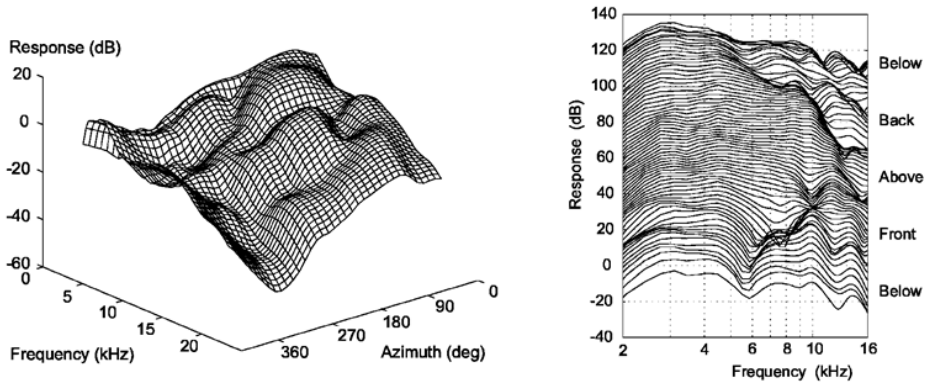


Figure 1.2: Log-magnitude HRTFs are smooth along spatial and frequency domains (horizontal and vertical plane directions are shown).

mounted on a rotatable hoop; the loudspeaker must be able to play a test signal upto the upper bound of the hearing range ($16 - 20$ kHz) and the mic must be able to sample at twice that rate. The equipment setup should limit the distortions caused by its own apparatuses via sound-absorption/padding. Moreover, the recordings are typically made in an near an-echoic chamber as to avoid reflections off the ground and walls. Second, the subject must commit a considerable investment of time and will-power to the measurement process. Measurement time is not instantaneous due to the speed of sound and must also factor in changes in the position and number of loudspeakers. The subject's head and torso must also remain immobile as to maintain consistency with the spherical coordinate system. These restrictions have motivated several alternative means for acquiring or indirectly inferring HRTFs. While such methods that circumvent the direct measurement process are prone to be less accurate, their varying accessibility to the general public may be worth consideration. We provide a deconstruction as follows:

One categorization of the various HRTF measurement processes is to consider the roles between the human subject and the machine-leaner (measurement methodology)

during the acquisition process. A role may be evaluated in terms of the amount of work, its computational costs, its accuracy, and its time to completion (see Table 1.1). For a point of reference, the roles of the subject and machine-leaner in the direct measurement process are presented: The subject's role is passive as the two microphones placed within the ear canals act as the listener/receiver of the test signal. The machine-learner is passive in the sense that the raw microphone recordings need little post-processing following Eq. 1.1 to derive the HRTF representation. The post-processed HRTFs are accorded the highest accuracy. The measurement time varies according to the number of loudspeakers. The monetary costs tend toward the high-end range depending on the design of the apparatus.

Method	Human-Listener	Machine-Learner	Accuracy	Equipment Costs	Measure. Time
Direct Measurements	Passive	Passive	High	$$-$$$	Varies
Boundary Elements	Passive	Active	Moderate-High	$$$	Low
Inference by Anthropometry	Passive	Active	Low-Moderate	$	Low-Moderate
Listening Test + Hand Tuning	Active	Passive	Low-Moderate	$	High
Listening Test + Recommender	Active	Active	Moderate	$	Moderate

Table 1.1: Functional and structural cost analysis of various HRTF acquisition methods.

Reciprocal direct measurements: The direct measurement process can be accel-

erated in time via the acoustical principle of reciprocity [26]. In this setup, the speaker and microphone positions are swapped so that multiple microphones positioned along a spherical grid centered over a subject can simultaneously record a single impulse or test signal originating from a microspeaker that is inserted into a subject's blocked ear canal. This significantly decreases measurement time as the impulse responses can be recorded in parallel (simultaneously) w.r.t. two serial impulses originating at the left and right ears canals; the number of impulses is no longer a function of the number of measurement directions and the total waiting time is reduced to that of the attenuation of two impulse trains. The trade-off is the increased costs of number of microphones and the need for a specialized wide-band microspeaker that fits into the ear-canal.

Boundary elements methods: It is possible to simulate the acoustic scattering of sound off the listener's anthropometry provided that one has a high-resolution discretization of the surface or boundary conditions. The acoustic scattering process obeys the 3D Helmholtz equation (see section 1.3) subject to boundary conditions that arise from the presence of a surface whose sound field is being computed. For HRTFs, such a surface is defined by a discretized mesh of the subject's head, torso, and outer ears, often generated from a point cloud measured by a laser scanner; similar to the direct measurement process, the subject's role is passive. The surface polygons or mesh may be generated from computational triangulations algorithms (e.g. Delaunay) of the point cloud in post-processing. The solutions to the Helmholtz equation are the HRTFs at varying frequencies, which are found via boundary element methods (BEMs). The simulated HRTFs are sensitive to the resolution of the mesh discretization where the Nyquist theorem establishes a minimum of two samples per the shortest wavelength or highest frequency of interest.

In practice, an additional $6 - 10$ samples are required [37] due to numerical imprecision and convergence issues for solving large system of equations in the BEM. Approximation methods such as conjugate gradients accelerated by the fast multipole method are often employed. The sample resolution limits the accuracy of the simulated solutions; for the full human auditory range (upto $16 - 20$ kHz), 3D scanning technologies must capture boundary elements less than 3 mm. Cheaper photo imaging technologies (point clouds from cameras, depth maps from Microsoft Kinect) do not satisfy the resolution requirements and have difficulty capturing regions of concavity in the outer-ear. The faster measurement time for the subject via scanning technologies is shifted onto greater computational work performed by the machine-learner.

Inference by anthropometry: A further relaxation of the BEM for simulating HRTFs correlates an high-level representation of the subject's mesh with the HRTFs via computational methods. The high-level representation is the coarse set of biometrics (physically measured anthropometry) of a subject's head, torso, and pinna that are either measured by hand or inferred from calibrated photo-images; similar to the direct-measurement process, the subject remains passive as another individual performs the measurements or a photograph is taken. For a dataset of corresponding pairs of anthropometry features and HRTFs that belong to the same subjects, the two domains can be related via regression models such as multiple linear regression, support vector regression, Gaussian process regression, and neural network models [31, 32]. This removes the need of an expensive scanning technologies and a time-consuming BEM solver at a greater cost of accuracy. In practice, most of these regression models are over-fitted (large generalization error) due to a small sample size of available subject data for training as existing datasets

contain less than 50 subjects. Combining multiple datasets is difficult as studies have shown that HRTFs belonging to the same subject but measured by different labs exhibit large variances [19].

Listening tests and HRTF query-selection: In a sound-source localization test, the subject listens to a test signal constructed from filtering a Gaussian white noise process with a query HRTF over a pair of headphones (see section 1.4); localization of the query-HRTF by the subject is articulated by reporting the sound-source direction in spherical coordinates, often through a GUI. The more difficult "query-selection" task (determining which HRTF for the subject to localize as to match a target-direction) is the subject of several works in literature. Such methods are inexpensive (equipment-wise) as it boot-straps or reuses devices (headphone, GUI) that the public possess. Instead, the costs are deferred to the human-listener and machine-learner who must learn the non-linear relationship between HRTF and sound-source direction.

In hand-selection/tuning methods, the listener either selects the query-HRTF out of a large candidate dataset or adjusts the HRTF magnitude spectra over a graphical user interface (GUI). Hand-selection methods require significant time due to the large number of candidate HRTFs. Hand-tuning methods (speed and accuracy) depend on the abilities of the listener; expert listeners can directly adjust the frequency components of the min-phase magnitude HRTF spectrum in Eq. 1.2 with moderate precision to reflect changes in spatialization [33]. A non-expert can be given a low-dimensional HRTF representation such as the leading principal components for adjustment. Another factor is the number and spacing of target-directions over the spherical coordinate domain; nearby directions are expected to share similar HRTFs.

In machine-selection methods, the learning task by the listener is reassigned to a learning algorithm which possess both prior knowledge of HRTFs are distributed w.r.t. measurement directions and posterior knowledge of all previous query-HRTF to localized directions made by the user. Two variations are posed: The first is a ranking problem where the listener orders a small set of candidate HRTFs according to the quality of spatializations and the distances from the target direction [34]; HRTFs over subsequent rounds are adjusted according to a genetic algorithm. One drawback is that the rankings only apply to fixed set of target directions and thus requiring restarts for new directions. The second variation is a blind-recommendation scheme where the listener localizes an HRTF without knowledge of the target directions. The learning algorithm has knowledge of a set of target directions and is able to compute and minimize an error distance for future queries as a search/optimization problem [20]. Developing a robust learning algorithm for this task remains an open challenge; several works in this dissertation address this problem.

1.3 Sound-Fields and Spherical Interpolants

While any finite collection of HRTFs can be physically measured, learning a continuous representation over the spherical coordinate domain is more useful. This has practical relevance as the collection of measured HRTF directions may be sparse and non-uniformly distributed over the spherical coordinate domain depending on the acquisition method. For direct measurements, some HRTFs that would have belonged near the bottom of the head are missing due to the in-feasibility of placing the measurement apparatus along

those directions. Measurement techniques based on listening tests only learn a sparse set of HRTFs due to time constraints. Moreover, VADs need to render sound-sources along any direction which necessitates having a continuous approximation of the entire sound-field. We refer to such approximations as "spherical interpolants" which have representations expressible in terms of a truncated spherical harmonic basis expansion given by

$$\begin{aligned} f(\theta,\phi) &= \sum_{n=0}^{p}\sum_{m=-n}^{n} f_n^m Y_n^m(\theta,\phi), \\ Y_n^m(\theta,\phi) &= (-1)^m \sqrt{\frac{2n+1}{4\pi}\frac{(n-|m|)!}{(n+|m|)!}} P_n^{|m|}(\cos\theta) e^{im\phi}, \end{aligned} \tag{1.3}$$

for truncation number p, the orthonormal spherical harmonics $Y_n^m(\theta,\phi)$, and the associated Legendre polynomials $P_n^{|m|}(\cos\theta)$. This use of the spherical harmonic basis for HRTFs is justified in part by models of wave propagation from the domain of computational physics [38, 39].

For example, the Helmholtz wave equation over the 3D sound field that is separable in frequency and space [40] is given by

$$\nabla^2\psi(k,r) + k^2\psi(k,r) = 0, \tag{1.4}$$

for spatial frequency (wavenumber) $k = \omega/c$, the speed of sound c, and the Fourier transform of the pressure $\psi(k,r)$. The solutions to the Helmholtz equation are expressed

as a series of regular and singular spherical basis functions $\psi = \psi_{in} + \psi_s$ given by

$$\begin{aligned} \psi_{in}(k,r) &= \sum_{n=0}^{\infty} \sum_{m=-n}^{n} A_n^m R_n^m(k,r), \quad \psi_s(k,r) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} B_n^m S_n^m(k,r), \\ R_n^m(k,r) &= j_n(kr) Y_n^m(\theta,\phi), \quad S_n^m(k,r) = h_n(kr) Y_n^m(\theta,\phi), \end{aligned} \tag{1.5}$$

where A_n^m, B_n^m are the weights, and $j_n(kr)$, $h_n(kr)$ the spherical Bessel and Hankel functions of the first kind respectively. The expansions of ψ in Eqs. 1.4, 1.5 are typically truncated to $n \leq N$ terms as a function of wavenumber $N = ka$, where a is the scatter radius.

An open challenge is to find such spherical interpolants that generalize the entire sound field from only a few measurements; several works are described in literature but have a number of shortcomings. It is possible to learn a least squares fitting of a truncated spherical harmonic basis to the per-frequency HRTFs [41]. However, such a method suffers from poor conditioning of the basis matrix due to non-uniform distributions of measured directions (e.g. randomized measurement directions in cross-validation testing); one solution is to regularize the matrix problem (e.g. Tikhonov regularization) or improve the condition number through the truncation of small singular values. Another family of method uses spherical spline interpolations [42, 43] for the non-parametric fitting of per-frequency HRTF measurements; the splines consist of a Legendre polynomial basis over the sphere. Unfortunately, choosing the model-order (smoothness terms and truncation number) for these bases is not automatic; such methods also ignore the observation that correlated measurements along the frequency domain can be used to further reduce model-order. Several works in this dissertation address these concerns.

1.4 Spatial Audio Synthesis and Playback

The final step in the spatial audio design is the synthesis of an arbitrary sound-source with one or more HRTFs for playback (see Figure. 1.3). For a single minimum-phase HRIR filter that characterizes the linear time-invariant system of the sound-path from sound-source to ear canal along directions θ, ϕ, and an input mono-channel sound-source x, the output signal is computed via the convolution operation $*$ given by

$$x_l = x * \hat{h}_{l,\theta,\phi}, \quad x_r = x * \hat{h}_{r,\theta,\phi}, \tag{1.6}$$

for time sample i, and left and right minimum-phase impulse responses $\hat{h}_{l,\theta,\phi}$, $\hat{h}_{r,\theta,\phi}$. The discrete time-domain convolution between arbitrary signals u, v of lengths $|u|$, $|v|$ in $w = u * v$ is given by

$$w(i) = \sum_j u(j)v(i-j+1), \tag{1.7}$$

where $u(j) = 0$ for $|u| < j < 1$ and $v(i-j+1) = 0$ for $|v| < i-j+1 < 1$. The computational cost of the direct convolution is thus quadratic $\mathrm{O}(|u||v|)$ operations. Digital signal processing methods have asymptotically lowered this cost by transforming both filters into the frequency domain (Fourier basis); discrete convolution via fast Fourier transform operations $\mathcal{F}\{\}$ [44] is given by

$$w = \mathcal{F}^{-1}\left\{\mathcal{F}\left\{u\right\} \circ \mathcal{F}\left\{v\right\}\right\}, \tag{1.8}$$

where $\circ$ is the Hadamard or element-wise product and requires O $((|u| + |v|) \log (|u| + |v|))$ operations. The outputs $x_l(i)$, $x_r(i)$ are simultaneously played over left and right ear headphone channels respectively with appropriate time-delay.

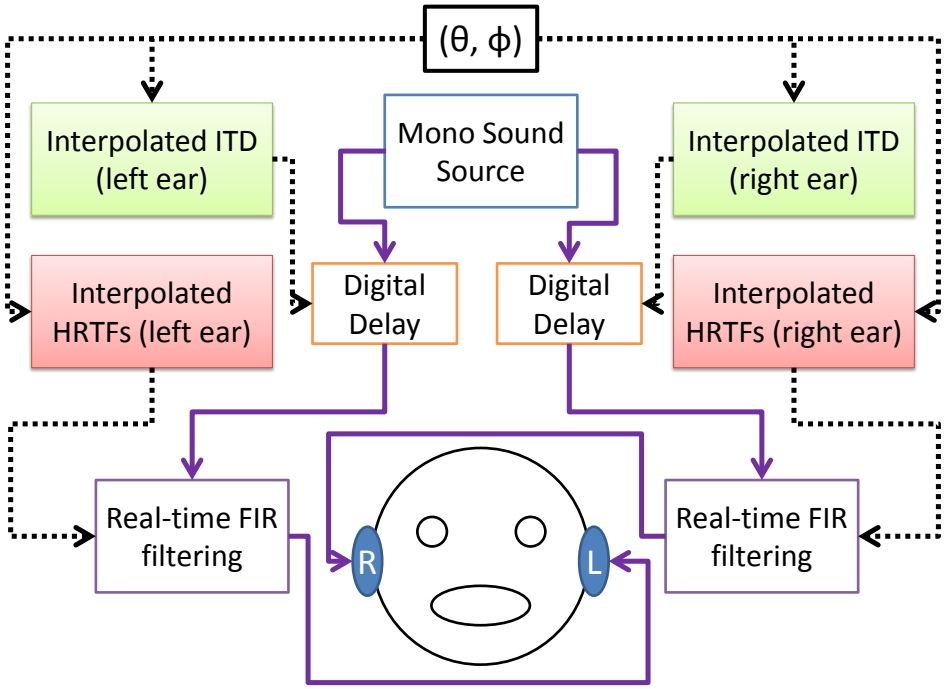


Figure 1.3: Playback along (θ, ϕ) via interpolated min-phase HRTFs with time-delay.

A number of challenges are present in this rendering stage which are motivated by a need for low-latency processing and real-time filtering. If the interpolated HRTFs are generated on the fly, then the number and forms of computations must be low and inexpensive respectively. Interpolations over a sphere tend to require evaluations of transcendental (e.g. exponential) and special (e.g. Legendre) functions that are not suited (slow) for some hardware; many boards have dedicated processors for such operations. One alternative is to pre-compute and store the interpolated HRTFs over an arbitrarily dense spherical coordinate grid. Run-time interpolation would follow nearest-neighbor search techniques over a look-up table. A second problem is the choice between time

and frequency domain convolution between HRTF and input sound-source signals. It is well known that both methods have a cross-over point in terms of their computational costs w.r.t. the filter lengths; time-domain convolution is asymptotically slower than the frequency-domain variants but the latter has an overhead cost. If the HRIR filters can be made short and sparse, then the time-domain convolution will be faster; several works in this dissertation address these concerns.

1.5 Machine Learning

The field of machine learning is comprised of several sub-fields that are concerned with the prediction and knowledge discovery of data independent of an expert-domain. We introduce several algorithms belonging to the sub-fields of supervised and unsupervised learning without the loss of generality to domains outside spatial audio. The choice of these algorithms is motivated by their ability to make meaningful inferences between large/varied collections of data and their potential for computationally efficient real-time processing. The former is accomplished by adapting domain-specific assumptions into their designs. The latter is accomplished via fast numerical and linear algebra techniques.

Supervised-learning algorithms attempt to map inputs to labeled (output) data which are already known in a training set. A number of *non-parametric* methods that belong to this class do not have fixed model-orders (parameter sizes) but instead adapt in complexity to the number training inputs that are used for inference. Popular *kernel* methods such as support vector machines (SVMs) [45] and Gaussian process (GP) regression (GPR) [46] have parameters that scale in size w.r.t. the evidence set (number of samples

conditioned upon). This property is useful for making inferences between domains such as HRTFs, measurement/sound-source localization directions, and human anthropometry without having knowledge of their causal relationships. However, non-parametric methods are computationally expensive in both memory and time which are problems that we address in our works. Related are semi-supervised active-learning methods which acquire labeled data by interactively querying the user. We develop some of these methods for the problem of recommending/learning HRTFs for the listener to localize via listening tests.

Unsupervised-learning algorithms attempt to discover an underlying structure behind unlabeled data. Clustering methods such as k-means [47] are related to non-negative factorization methods [48, 49] which decomposes data into additive parts of the whole. Low-dimensional generative models such as mixture of Gaussians [50, 51], auto-encoder neural networks [52, 53] are able to both encode data into high-level features and decode the features back into the original space. These methods are useful for the structural decomposition of collections of HRTFs where methods for sampling and searching along these low-dimensional representations are more efficient.

1.6 Organization

A brief summary of each chapter and their contributions are presented below:

Chapter 2: We present a GP model for localizing arbitrary sound-sources by a human listener using prior knowledge of his/her HRTFs. The source-signal's contents are removed via cancellation of the left and right audio streams; the resulting features are expressed as various ratios of the underlying HRTFs which are known a priori; the

features are used as predictors of the sound-source directions in GPR models. Two further problems are proposed: First, we show how only a small-subset of features are relevant for human-accurate localization over the entire spherical coordinate domain via greedy forward selection methods [54]. Second, we formalize the HRTF recommendation system into an *active-learning* [55] problem based on the application of GP localization models for the global optimization of smooth functions [56]. Experiments with human and virtual listeners show that the learned HRTFs are localized closer to their targeted directions than non-individualized HRTF guesses.

Chapter 3: We present an efficient joint spatial-frequency covariance model for HRTF interpolation via GPR. Model-order selection, generalization error, and computational concerns from section 1.3 are addressed: The GP model exploits a "gridded" structure between the HRTF spherical-frequency input domains which allows the covariance/Gram matrix to be factorized into Kronecker product matrices [57]. Asymptotic complexity reductions from best-case cubic to linear runtime and quadratic to linear space costs are obtained. The model generalizes to arbitrary input dimensions and for sparse-GPR [58] methods. Extensions to HRTF spectral extrema extraction, treatment of missing/extra data, efficient feature subset-selection, and fast covariance function evaluations via series expansions are made.

Chapter 4: We present a method for "fusing" same-subject HRTF datasets collected by separate labs over different measurement grids in a GP setting based on chapter 3. This is motivated by a need for larger training datasets and the observation that same-subject HRTFs collected by different labs exhibit large variations. A data fusion metric using GP log-marginal likelihoods is derived. Two data transformations that capture the

inter-lab variations are learned which allow for inter-lab HRTFs belonging to different subjects to be compared.

Chapter 5: We present an efficient numerical method for relating left and right ear HRIRs in terms of time-delayed reflections based on a non-negative least squares (NNLS) formulation [59] of a linear Toeplitz system of equations. The NNLS method is an active-set variable selection method that we accelerate via efficient QR matrix updating and downdating operations. The algorithm is then parallelized for multi-core architectures (CPU and GPU) using OpenMP [60] and CUDA [61].

Chapter 6: We present a fast convolution method based on a sparse representation of HRIRs motivated in section 1.4. We extend a well-known non-negative matrix factorization method [49] to Toeplitz constrained matrices where a collection of HRIRs that belong to the same subject is factorized into convolutions between direction-independent and direction-dependent components. The latter direction-dependent components are non-negative and can be tuned for sparsity at a cost of reconstruction accuracy of the original HRIR. Convolutions between arbitrary signals and our HRIR representation in the time-domain are shown to be more efficient than fast Fourier transform (FFT) based convolutions.

Chapter 7: Conclusions are made and several open problems are discussed.

Chapter 2: Gaussian Process Models for Sound-Source Localization and Active-Learning

2.1 Introduction

Many animals possess a remarkable omnidirectional sound localization ability enabled by subconsciously processing subtle features in the sounds received at the two ears from a common source location. For humans, these features arise due to the incoming acoustic wave scattering off the listener's anatomical features (head, torso, pinnae) before reaching the eardrum. The spectral ratio between the sounds recorded at the eardrum and that would have been obtained at the center of the head in absence of the listener is known as the head-related transfer function (HRTF) [7]; HRTFs are thus specific to the individual's anthropometry, wave direction, and contain other important cues such as the interaural time delay (ITD) and the interaural level difference (ILD) [24]. Moreover, knowledge of individualized HRTFs allow for perceptually accurate 3D spatial audio synthesis [62–64].

We investigate the *pre-image* problem, namely how pairs of left and right ear HRTFs and functions of HRTFs (features based on them) map back to their measurement directions. This is related to the problem of sound-source localization (SSL) where under simple (anechoic) conditions, the direction of an acoustic event can be inferred from

multi-receiver recordings of the sound spectrum by expressing the spectral cues solely in terms of the receiver's transfer functions (independent of their actual content). This is of interest in robot perception (e.g. for event detection and localization [65, 66]), where the receiver's transfer functions can be measured beforehand. For humans, this problem is restricted to two receivers (human ears) where functions of left and right pairs of HRTFs are mapped to their measurement directions in place of SSL directions. Thus, it possible to model this relation as either a classification or a regression problem between the two domains. Many works in literature have attempted similar tasks.

2.1.1 Prior Works

Cue-mapping [67] uses ITD, ILD, and interaural envelope difference features paired with azimuth directions in a weighted kernel nearest-neighbor (NN) setting. A linear mapping between ITD, ILD, and HRTF notch frequency features to spherical coordinates can be learned [65]. A self-organizing map between input ITD, spectral notches features and output horizontal and median plane coordinates can be trained [68]. Conditional probability maps derived from per-frequency ITD and ILD can be used to estimate direction via a maximum a posteriori estimator [69]. A probabilistic affine regression model between interaural transfer functions and the direction is possible [70].

Most closely related to our work are the source-cancellation and match-filtering algorithms [71–74], where the binaural recordings (S_L left, S_R right ears) are represented as convolutions of a common sound-source signal S and the appropriate filters; for recording done in an anechoic space, these filters are the same-direction HRTFs (H_L left, H_R right

ears). The per-frequency domain representation is given by

$$S_L = H_L \circ S, \quad S_R = H_R \circ S, \tag{2.1}$$

where $\circ$ is element-wise product. The source-signal S is removed by computing the ratio between left and right channel recordings ($\frac{S_L}{S_R} = \frac{H_L}{H_R}$). These binaural features, which are reduced to ratios of HRTFs, can be compared to those pre-computed from the subject's collection of measured HRTFs; the measurement direction belonging to the maximally cross-correlated pair is reported as the sound-source direction. Such an approach can be interpreted as a nearest neighbor (NN) classifier where the binaural features and measurement directions are single class instances and labels respectively.

2.1.2 Present Work

We propose a generalization of the match-filtering algorithm that addresses several deficiencies: While an NN classifier is accurate for a large number of training samples, it does not report out-of-sample spatial directions unless specified in a regression context. Linear regression methods via ordinary least squares (OLS) regressors[1] often perform poorly due to inaccurate assumptions on the model complexity (number of parameters) and the linearity between predictors and outputs. Common issues include over-fitting the model to noise that arise from parametric OLS methods and under-fitting the training data from assumptions of linearity. Instead, we adopt a non-linear and non-parametric[2] Gaussian

[1] $\mathbf{y} = \mathbf{x}^T\beta, \quad \beta = \left(X^T X\right)^{-1} X^T Y$, for parameters β

[2] Number of parameters is proportional to the number of data samples conditioned upon for inference.

process (GP) regression (GPR) [46] framework to address these issues.

GPR is a *kernel method*[3] that places weak assumptions on the joint probability distribution[4] of *latent function realizations* that would model the output observations (spatial directions) in a Bayesian setting. Observations are drawn (realized) from a high-dimensional normal distribution that represents the joint probability density function of a collection of random variables indexed by their predictor variables. GPs have several attractive properties that are well-suited for SSL.

Based on the observation that HRTFs corresponding to different spatial directions covary smoothly with the considered binaural features (see sections 2.3), we show they can be modeled via simple stationary GP covariance functions (see section 2.4). The GP Bayesian formulation allows for the choice of the covariance function, which governs the smoothness between realizations at nearby predictors, to be automatically selected by evaluating a data marginal-likelihood criterion (goodness-of-fit); covariance functions belong to a function class and are specified by their "hyperparameters" (parameters that describe distributions). This allows the covariance function hyperparameters to be learned without the need for cross-validation and provides insights as to the intrinsic dimensionality of the high-dimensional feature space that the binaural features are mapped to. Most importantly, uncertainties in GP prediction are well-defined in terms of both prior and posterior distributions; the predicted variances at different inputs are tractable. Thus, GPR generalizes NN classifiers as it makes non-linear inferences to observations outside the training set. By the representer theorem, kernel methods such as support vector regression

[3]Predictor variables are implicitly mapped to a reproducing kernel Hilbert space whose inner products are taken to be evaluations of a valid Mercer kernel or covariance function.

[4]Normal distribution defined by prior mean and covariance functions of predictor variables (binaural features).

(SVR) [45] and GPR make predictions expressible as linear combinations of non-linear covariance evaluations between the training features/observations and the test features.

In general, GPs perform better (make accurate inferences) with more observations (data) than other non-linear regression methods that do not encode and select for prior data-assumptions. The trade-off is its high computational costs ($\mathrm{O}\left(N^3\right)$ operations for N number of observations) for both model-selection and inference; scaling GPs for for large datasets is an active field of research. Fortunately, the availability of high quality datasets, computational resources, and faster algorithmic formulations have allowed us to overcome these problems. In previous works, we have used several properties of HRTF datasets to to perform fast GP based HRTF interpolation [18] and data-fusion [19]. The current work is a major extension of our recent work on binaural SSL [21]. For future references, we refer to GPs that predict SSL directions as **GP-SSL** models (see section 2.4 for a complete derivation).

2.2 Formulation of Problems

This work investigates two problems related to GP-SSL models (see Fig. 2.1). For notation, we refer to a binaural feature as a D-dimensional vector $\mathbf{x} \in \mathbb{R}^D$ (D is number of frequency bins), the measurement direction as the unit vector $\mathbf{y} \in \mathbb{R}^M$ ($M = 3$ for the standard Cartesian basis), and collections of the aforementioned quantities (N number of samples) as concatenated into matrices $X \in \mathbb{R}^{N \times D}$ and $Y \in \mathbb{R}^{N \times M}$. The binaural features are independent of the sound-source content and thus strictly functions of the subject's HRTFs (see section 2.3). GP-SSL models are thereby specified and trained over

known HRTFs and measurement directions belonging to CIPIC [1] database subjects.

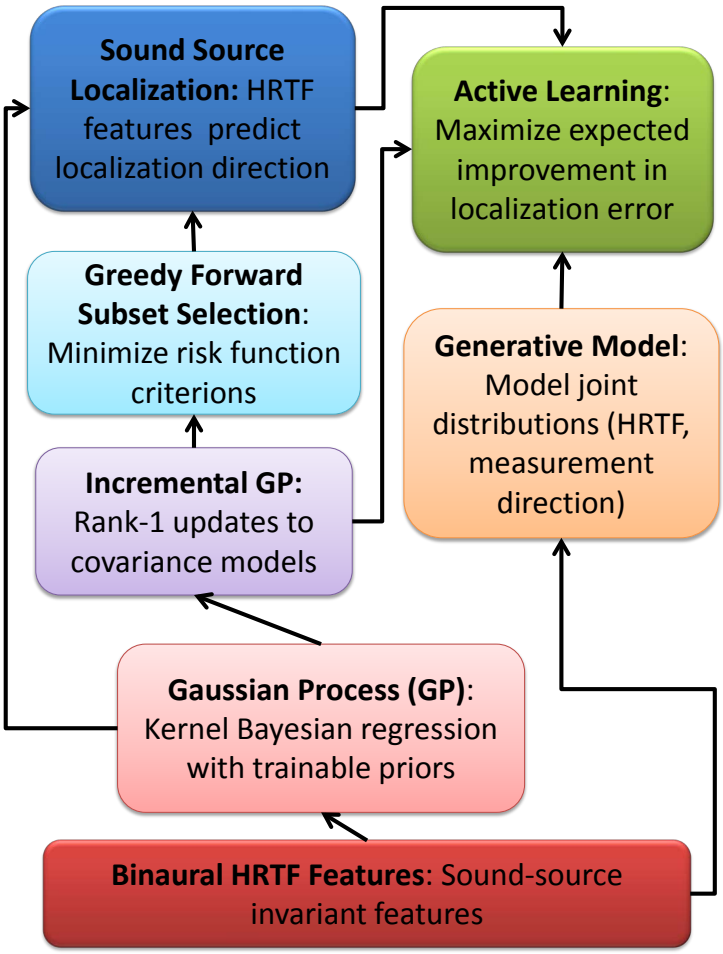


Figure 2.1: Gaussian Process Regression with binaural features (bottom two boxes) to perform two types of inferences. On the left are shown the steps needed to perform sound-source localization. On the right is shown an active-learning framework that combines SSL with listening tests to learn a listener's HRTFs.

2.2.1 Feature subset-selection

Subset-selection for non-parametric methods such as NN and GPR is an important technique for reducing the model-order complexity and run-time costs for inference. SSL models that are trained with randomized subsets of samples trade measurement and prediction costs for localization accuracy. Increasing the density of measurement samples

over the spherical grid results in a linear increase to both NN classification computational cost and accuracy, a quadratic and cubic increase to respective GP inference and training computational costs, and a non-linear increase to GP localization accuracy. We show how GP-SSL models using small and non-uniform subset-selected samples (which are most informative) make more accurate predictions over the full spherical grid than models evidenced on a randomized subset.

A simple greedy forward-selection (GFS) algorithm [54] that sequentially incorporates training samples into a subset without considerations in future iterations is implemented. It ranks all training samples outside the subset via a user-defined objective function (risk function) and adds the minimizer into the subset. We propose a class of risk functions that generalizes the GP prediction errors and show that the subset-selected GP-SSL models localize directions more accurately than models evidenced on randomized inputs (see section 2.5); only a small fraction of training samples are required for reasonable accuracy (5°).

2.2.2 Active-learning for individualizing HRTFs

Individualized HRTFs are needed for synthesizing accurate spatial audio that resolve front-back and up-down directional confusion [62–64]. Due to the difficulties of directly measuring HRTFs [30], a number of works have sought indirect means for learning the subject's HRTFs: regression models between the individual's physically measured anthropometry and his/her HRTFs can be learned via neural-network [32] and multiple non-linear regression models [31] but do not generalize well to test subjects. HRTFs can also

be learned through listening tests [33,75] by having an individual listen to a query HRTF $\mathbf{x}$ convolved with white Gaussian noise (WGN) (heard over a pair of headphones), localize the test signal (report a direction $\mathbf{v} \in \mathbb{R}^3$), and then hand-tune the spectra of $\mathbf{x}$ or choose a new $\mathbf{x}$ out of a large candidate pool over a graphical user interface (GUI) as to move subsequent localizations towards a target direction $\mathbf{u} \in \mathbb{R}^3$. The hand-tuning/selection step can be replaced by developing a recommendation system that selects for the query HRTF between rounds (steps) of localization. The listener can rank candidate HRTFs chosen from a genetic algorithm[5] [34]. HRTFs can also be tuned along a low-dimensional autoencoder space [20] where $\mathbf{u}$ is unknown to the listener.

We propose to formulate the recommendation problem in an *active-learning* [55] context described as follows: given a finite set of candidate HRTFs X^C sampled from a prior distribution (database or generative model), determine the HRTF from the X^C that the listener would localize nearest to $\mathbf{u}$ within T rounds of localizations. During round $t \leq T$, the recommender selects a query $\mathbf{x}$ that the listener labels as $\mathbf{v}_t(\mathbf{x})$ without knowledge of $\mathbf{u}$. The choice of $\mathbf{x}$ is referred to as the *query-selection* problem of minimizing the SSL error (SSLE) (modified cosine distance) given by

$$\mathbf{SSLE}(\mathbf{u}, \mathbf{v}_t(\mathbf{x})) = -\mathbf{u}^T \mathbf{v}_t(\mathbf{x}), \quad \arg\min_{\mathbf{x} \in X^C} \mathbf{SSLE}\left(\mathbf{u}, \mathbf{v}_t(\mathbf{x})\right). \tag{2.2}$$

Unfortunately, the minimizer in Eq. 2.2 is unlikely to be found within T rounds as X^C can be large and T must also be small as the cost of evaluating SSLE by the listener is high. It is more reasonable to model the SSLE function using an online regression

[5]Evaluates a fitness function w.r.t. localization accuracy of known $\mathbf{u}$

model (adapting HRTFs predictors of SSLEs after each round) and select for **x** based on two competing strategies: query-selection *exploits* the online model by choosing **x** that the model predicts will have low SSLE and *explores* **x** that has high model uncertainty in its prediction; both concepts are trade-offs that require probabilistic treatments of model predictions. Fortunately, GPs are well-suited to this task as all predictions are expressed as probabilistic realizations sampled from normal distributions. Thus, we propose to solve the modeling problem via **GP-SSLEs**[6], and the query-selection problem using a method of GPs for the global optimization of smooth functions [56, 76] (see section 2.6). The relation between these methods and the GP-SSL models is also shown.

2.3 Binaural Sound-Source Invariant Features

We consider several sound-source invariant features that can be extracted from short-time Fourier transforms of the left and right ear input channel streams in Eq. 2.1 (see Table 2.1 and Fig. 2.2); it is useful to express the discrete Fourier transformed signals by their magnitude and phase representations where $H(j\omega) = |H(j\omega)|\, e^{j\angle H(j\omega)}$. The features are expressed as ratios between left and right ear channel recordings that remove the effects of the acoustic content in S; the remainder is strictly a per-frequency function of same-direction left and right ear HRTFs derived as follows:

Table 2.1: HRTF sound-source invariant features X

$\log\left(\left\lvert\frac{S_L}{S_R}\right\rvert + 1\right) = \log\left(\frac{\lvert H_L\rvert}{\lvert H_R\rvert} + 1\right)$	Log-magnitude ratio
$\angle\frac{S_L}{S_R} = \angle H_L - \angle H_R$	Phase difference
$\frac{\lvert S_L\rvert}{0.5(\lvert S_L\rvert + \lvert S_R\rvert)} = \frac{2\lvert H_L\rvert}{\lvert H_L\rvert + \lvert H_R\rvert}$	Avg. magnitude ratio
$\{\lvert S_L\rvert, \lvert S_R\rvert\} = \{\lvert H_L\rvert, \lvert H_R\rvert\}$	Magnitude pairs for flat S

[6] GPs that predict the SSLE from HRTFs

Log-magnitude ratio (LMR) [71]: While the source-cancellation method removes the dependence on signal S, the resulting features are complex, noisy, and difficult to interpret. This can be avoided by considering the magnitude representation which gives the relative per-frequency energy between the channel signals. Adding a constant to the ratio prior to the log-transform penalizes the magnitude of the perturbation; adding a constant 1 constrains the log-transform to be non-negative.

Phase difference (PD): Similarly, the per-frequency phase of the complex channel signal ratio can be expressed by the phase-difference between left and right HRTFs. For identical S_L, S_R that differ by onset time-delays Δ_L, Δ_R, the phase-difference is simply the constant delay $\Delta_L - \Delta_R$ across all frequencies; this ITD can be related to azimuth angles via Woodworth's model [2]. For arbitrary S_L, S_R, the per-frequency phase-differences differ and are to be treated as independent variables in regression models.

Average magnitude ratio (AMR): The magnitude source-signal $|S|$ can also be removed by taking the ratio of left or right magnitude signals $|S_L|, |S_R|$ and the binaural average $(|S_L| + |S_R|)/2$. Without the constant factor, the feature can be interpreted as the per-frequency contribution of the left or right magnitude HRTFs to the additive binaural magnitude response. Unlike log-magnitude ratio features that approaches a singularity as $|H_R| \to 0$, these features are bounded in the interval $[0, 2)$ and finite everywhere unless the binaural average is zero.

Magnitude pairs (MP): The magnitude pairs are the concatenation of the original left and right magnitude HRTFs that could be derived from convolution with a WGN S with zero mean and unit variance. The power spectrum of $|S|^2$ is constant across all frequencies and so $|S_L|, |S_R|$ would be constant factors of magnitude HRTFs. Such

conditions arise during listening tests where the source-signal S can be specified; the test features can then be derived from per-frequency division given by $H_L = S_L/S$ and $H_R = S_R/S$.

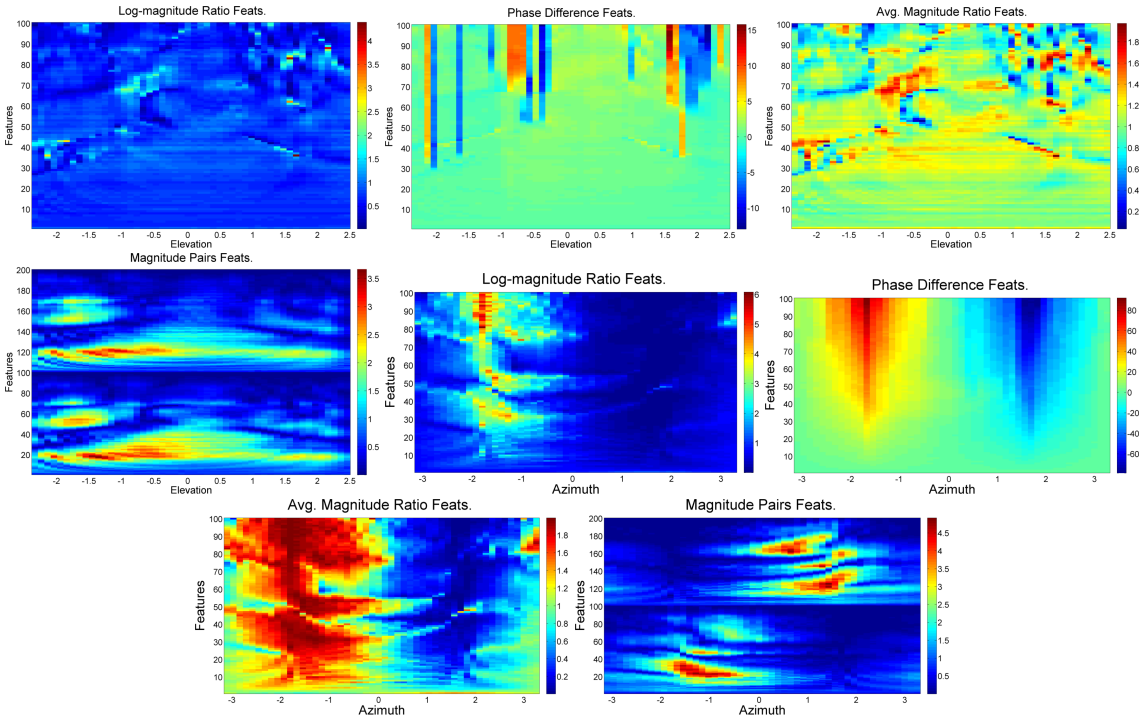


Figure 2.2: Binaural features extracted from CIPIC subject 156 HRTFs are shown for horizontal and median plane directions.

2.4 Gaussian Process Regression for SSL

In a general regression problem, one predicts a scalar target variable y from an input vector $\mathbf{x}$ of independent variables based on a collection of available observations. A common Bayesian approach for inference assumes that the observation y is generated (realized) from a latent function $f(\mathbf{x})$ given by

$$y = f(\mathbf{x}) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2), \tag{2.3}$$

which is corrupted by additive Gaussian white noise with zero mean and constant variance σ^2. This latent function is given the form of a kernel regression $f(\mathbf{x}) = \phi(\mathbf{x})^T\beta, \quad \beta \sim \mathcal{N}(0, \Sigma_p)$ where the function $\phi(\mathbf{x}) : \mathbb{R}^D \to \mathbb{R}^{D^*}$ maps the inputs $\mathbf{x}$ into a high-dimensional space before computing the inner product with a vector of parameters realized from a collection of random variables with a prior multivariate normal distribution $\beta \in \mathbb{R}^{D^*}$. Unlike linear regression, the parameters β are not explicitly found in order to perform inference but are marginalized in order to compute the first two moments (mean and covariance) of function $f(\mathbf{x})$ given by

$$\begin{aligned} \mathbb{E}\left(f(\mathbf{x})\right) &= \phi(\mathbf{x})^T\mathbb{E}\left(\beta\right) = 0, \\ \mathbb{E}\left(f(\mathbf{x})f(\mathbf{x}')\right) &= \phi(\mathbf{x})^T\mathbb{E}\left(\beta\beta^T\right)\phi(\mathbf{x}') = \phi(\mathbf{x})^T\Sigma_p\phi(\mathbf{x}'). \end{aligned} \tag{2.4}$$

The latent function realizations $f(\mathbf{x})$ are thus drawn from a multivariate normal distribution with mean $\mu(\mathbf{x}) = 0$ and variance $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T\Sigma_p\phi(\mathbf{x}')$. For $\Sigma_p = I$, the inner product can be replaced with the covariance function $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T\phi(\mathbf{x}')$ which GPs generalize as follows:

A GP f is a collection of random variables where any finite subset indexed at N inputs $X = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ has the joint multivariate normal distribution given by

$$[f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)] \sim \mathcal{N}(\mu(X), K(X, X)), \tag{2.5}$$

and thus fully defined by the prior mean function $\mu(\mathbf{x})$ and the prior covariance function $k(\mathbf{x}, \mathbf{x}')$. The prior mean function and vector $\mu(X) \in \mathbb{R}^N$ are set to zero without loss of generality following Eq. 2.4. The covariance (Gram) matrix $K(X, X) \in \mathbb{R}^{N\times N}$ is

characterized by the pairwise covariance function evaluations $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$; the covariance function is a positive semi-definite kernel (Mercer's condition) that establishes the existence of the eigenfunction $\phi(\mathbf{x})$. This allows kernel methods such as SVR and GPR to omit computing the exact mapping ϕ as the inner products in the high-dimensional space, representing the similarity measure between input features $\mathbf{x}, \mathbf{x}'$, are well-defined.

GP inference at test inputs $X_* \in \mathbb{R}^{N_* \times D}$ evidenced on training inputs X and the observations in $Y \in \mathbb{R}^N$ derives from the multivariate normal distribution of random variables $f_* = f(X_*)$ conditioned on $f(X) = Y, X$. This is given by

$$\begin{aligned} f_*|X, Y, X_* &\sim \mathcal{N}(\bar{f}_*, \bar{\Sigma}_*), \quad \bar{f}_* = K_{f*}^T \hat{K}^{-1} Y, \\ \bar{\Sigma}_* &= K_{**} - K_{f*}^T \hat{K}^{-1} K_{f*}, \end{aligned} \tag{2.6}$$

where $\hat{K} = K(X, X) + \sigma^2 I$ adjusts for the observation noise and $K_{f*} = K(X, X_*) \in \mathbb{R}^{N \times N_*}$ are pair-wise covariance evaluations between training and test inputs. We refer to the distribution in Eq. 2.6 as the posterior GP defined by the *posterior mean* and *posterior covariance* functions $\bar{f}_*$ and $\bar{\Sigma}_*$ respectively. The former represents the vector of expected outputs (prediction) at X_* and the latter is gives the confidence intervals (diagonal of the matrix) of the predictions.

For the GP-SSL model, X and $Y \in \mathbb{R}^{N \times 3}$ are the respective binaural features in Table. 2.1 and their measurement directions (unit vectors where $Y_i = Y_{:,i}$ are values along the i^{th} coordinate); test inputs X_* refer to the binaural features extracted from test signals. While it is possible to model all $M = 3$ output coordinates as a collection of M independent GPs $f_{1:M}(X) = \{f_1(X), \ldots f_M(X)\}$, a computationally cheaper alternative

is to specify a common prior mean and covariance function shared by all GPs. Specifying a shared covariance model between GPs is reasonable as the original HRTFs are originally measured over the same physical topology of a human subject from a near-uniform spherical grid of directions. Thus for inference, we use three independent GPs, with shared priors, to model left-right, front-back, and top-down coordinate directions by either sampling from their posterior distribution or reporting their posterior means.

2.4.1 Choice of Covariance Functions

The "smoothness/correlatedness" of realizations of $f(X)$ for similar X depends on the number of times that the covariance function is differentiable w.r.t. the input arguments. Consider the Matérn class of covariance functions where each function has varying orders of differentiation. For D-dimensional inputs, we can specify the GP covariance function as the product of D-independent Matérn covariance functions of identical class. Three common classes and the product covariance function are given as

$$\begin{aligned} K_{\frac{1}{2}}(r,\ell) &= e^{-\frac{r}{\ell}}, \quad K_{\frac{3}{2}}(r,\ell) = \left(1+\frac{\sqrt{3}r}{\ell}\right)e^{-\frac{\sqrt{3}r}{\ell}}, \\ K_{\infty}(r,\ell) &= e^{-\frac{r^2}{2\ell^2}}, \quad K(\mathbf{x},\mathbf{x}') = \alpha^2 \prod_{k=1}^{D} K_{\nu}(|\mathbf{x}_k - \mathbf{x}'_k|, \ell_k), \end{aligned} \tag{2.7}$$

for distance r and hyperparameters α, ℓ_k. Covariance functions K_ν are $\lfloor \nu \rfloor$ times differentiable and stationary due to their dependence on $|\mathbf{x}_k - \mathbf{x}'_k|$. Each function contains a length-scale or bandwidth hyperparameter ℓ_k that represents a distance in the domain $\mathbf{x}_k$ where outputs $f(\mathbf{x}_k)$ remain correlated; larger length-scales result in smoother f.

A general hyperparameter Θ is optimized by maximizing the data log-marginal

likelihood (LMH) of the observations Y given the GP prior distributions; the derivation follows from integrating over the realizations $f(X)$ by the product of data likelihoods (sampling Y from $f(X) + \epsilon$ and sampling $f(X)$ from the GP prior distribution). The LMH term $L = \log p(Y|X)$ and its partial derivative are both analytic and given by

$$\begin{aligned} L &= -\frac{M}{2}\left(\log|\hat{K}| + \frac{\mathbf{tr}\left(Y^T\hat{K}^{-1}Y\right)}{M} + N\log(2\pi)\right), \\ \frac{\partial L}{\partial \Theta_i} &= -\frac{M}{2}\left(\mathbf{tr}\left(\hat{K}^{-1}P\right) - \frac{\mathbf{tr}\left(Y^T\hat{K}^{-1}P\hat{K}^{-1}Y\right)}{M}\right), \end{aligned} \tag{2.8}$$

where $P = \partial\hat{K}/\partial\Theta$ is the matrix of partial derivatives. A larger LMH represents a better goodness-of-fit of the data to the GP prior mean and covariances assumptions. Moreover, different covariance functions with optimized hyperparameters can be compared in this respect without resorting to domain-specific metrics.

2.4.2 Model-Order and Cost Analysis

The GP model-order is proportional to the size of the GP prior distribution defined by the N-dimensional multivariate normal distribution in Eq. 2.5 (N is the number of training samples). The associated costs of both conditioning on the GP prior distribution for inference and performing hyperparameter training is dominated by the inversion of the Gram matrix (O (N^3) operations to compute and O (N^2) space to store). For large N, exact GP becomes intractable and most practitioners rely on randomized sampling techniques [77] to reduce the costs at the expense of accuracy. Two types of analyses for evaluating this trade-off are given: first, empirical cross-validation experiments can demonstrate how

data sampling (randomized and subset-selection) increases localization error. Second, the theoretical dimensionality of the feature space $\phi(\mathbf{x})$ in Eq. 2.3, despite not having been explicitly computed, can be estimated from an eigenanalysis of the GP Gram matrix. The distribution of eigenvalues (number of dominant ones) gives a minimum bound as to the number of input features whose mapping will contain most of the variances in the feature space.

To evaluate the dimensionality of $\phi(\mathbf{x})$, we refer to the method of kernel principal component analysis [78] of Gram matrix K. Its derivation expresses the eigenvectors v (principal directions) and eigenvalues λ (measure of variance captured by v) of the sample covariance matrix $\tilde{C}$ of features $\phi(\mathbf{x})$ in the high-dimensional space in the form of

$$\tilde{C} = \frac{1}{N}\sum_{i=1}^{N}\phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^T, \quad \tilde{C}v = \lambda v, \quad v = \frac{\sum_{i=1}^{N}\alpha_i\phi(\mathbf{x}_i)}{\lambda N}, \tag{2.9}$$

where $\alpha_i = \phi(\mathbf{x}_i)^T v$ are the component scores between the feature mapping and the eigenvector. Applying the "kernel" trick allows α to be reformulated in terms of the Gram matrix K as a tractable eigendecomposition problem given by

$$\begin{gathered}\sum_{j=1}^{N}\lambda\alpha_j = \sum_{j=1}^{N}\phi(\mathbf{x}_j)^T\tilde{C}v = \frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{N}\alpha_j K_{ij}, \\ K_{ij} = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j), \quad K\alpha = \lambda N\alpha,\end{gathered} \tag{2.10}$$

which finds the eigenvalues λ and scores α. Evaluating the contributions of the leading λ to the total energy $\sum_{i=1}^{N}\lambda_i$ estimates the number of eigenvectors that are relevant to $\phi(\mathbf{x})$.

2.4.3 Experiments

GP-SSL models (input binaural features LMR, PD, AMR, and MP from Table. 2.1 belonging to CIPIC subject 156) are trained (batch gradient descent of all covariance function hyperparameters ℓ_k via Eq. 2.8) for 50 iterations. For a domain-metric, we use the angular separation distance between two directions **u, u'** (predicted and reference directions) given by

$$\text{dist}\,(\mathbf{u}, \mathbf{u'}) = \cos^{-1} \frac{< \mathbf{u}, \mathbf{u'} >}{||\mathbf{u}||||\mathbf{u'}||}, \quad \mathbf{u}, \mathbf{u'} \in \mathbb{R}^3. \tag{2.11}$$

Goodness-of-fit: GP-SSL models are specified/trained on the full set of inputs X. The data LMHs in Table. 2.2 are computed for several covariance functions and feature types. The infinitely differentiable squared exponential K_∞ gives the best-fit (highest LMH) across all features (latent functions modeling the SSL directions are smooth w.r.t. changes in the feature space). This confirms the fact that a finite collection of HRTFs approximates a sound-pressure field that is continuous in space. The best-fitting binaural features are the MPs (WGN sound-source) and AMRs (arbitrary sound-source); the LMH gap between the two suggest that GP-SSL models will perform more accurately when the recorded magnitude spectra match that of the HRTFs. The LMH gap between AMR and LMR suggests that relative contribution may be a better indicator of SSL than relative intensities. The low LMH of PD models suggests that phase may not be useful for SSL over the entire spherical coordinate system.

Eigenanaylsis of K: The eigenvalues of the K are computed for GP-SSL models

Table 2.2: Data LMH for feature/GP covariance types

	LMR	PD	AMR	MP
K_∞	**2.69e+003**	**2.37e+003**	**3.9e+003**	**6.34e+003**
$K_{3/2}$	2.23e+003	1.5e+003	3.88e+003	6.29e+003
$K_{1/2}$	2.06e+003	460	2.24e+003	4.84e+003

trained/specified on the full dataset ($N = 1250$). Fig. 2.3 shows the contribution of the leading eigenvalues to the total energy; K_∞ specified by the four earlier features (LMR, PD, AMR, and MP) require respectively 150, 30, 100, and 15 leading eigenvectors to capture 90% of the total variance. The results suggest that feature mappings for MPs and PDs can be approximated with only a few samples while LMR and AMR feature mappings are more complex.

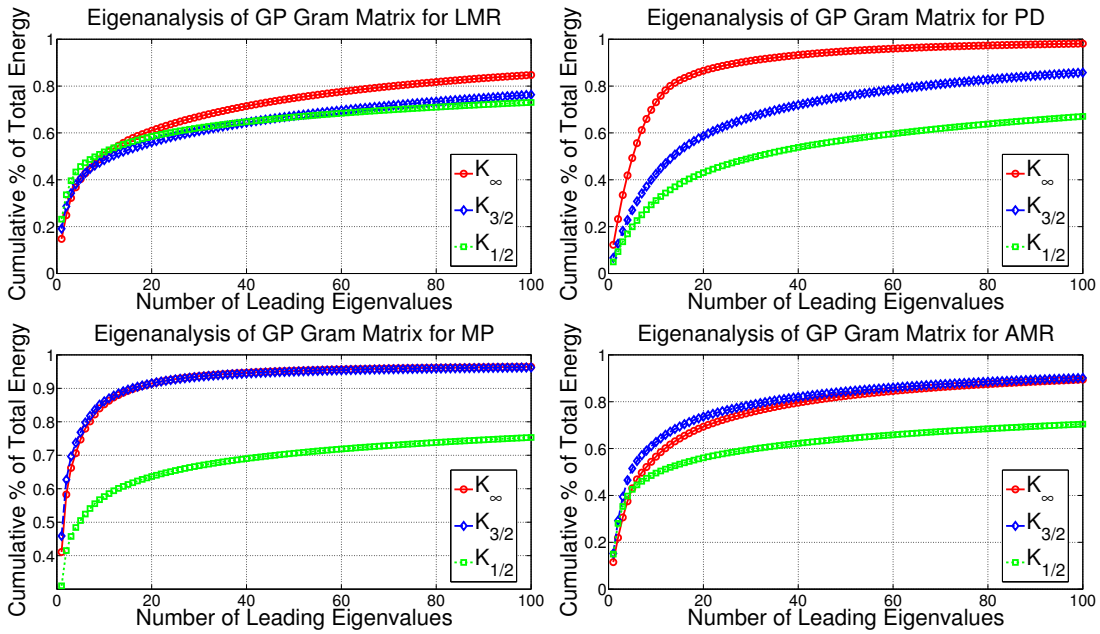


Figure 2.3: Cumulative energy of leading eigenvalues for K are shown for GP-SSL models (varying covariance functions and feature types).

Cross-validation: GP-SSL models are trained on a randomized third of the available feature-direction pairs ($N = 417$ out of 1250); inference follows Eq. 2.6 at all available inputs ($X_* = X$) where only the posterior mean directions are reported. Table

Table 2.3: Mean angular separation errors (degrees) for feature/methods

	LMR	PD	AMR	MP
OLS	29	27	22	5.4
NN	9.2	20	7.9	3.9
GP-SSL $K_{1/2}$	7.2	12	7	1.8
GP-SSL $K_{3/2}$	7.5	11	4.8	1.4
GP-SSL K_{∞}	**6.3**	**6.3**	**4.8**	**1.3**

2.3 shows the mean angular separation (Eq. 2.11) between predicted and reference directions for GP-SSL, NN classifier, OLS methods trained on the same data. Non-parametric methods (NN and GPR) outperform parametric methods (OLS) across all feature types. The MP and AMR features give the lowest errors across all methods (for a visual, see the first column of Fig. 2.4). OLS log-ratios perform the worse and suggest that the features are oversensitive linear predictors of change in localization direction. PD features, while useful for predictions on the horizontal plane, are insufficient for localizations over the full sphere.

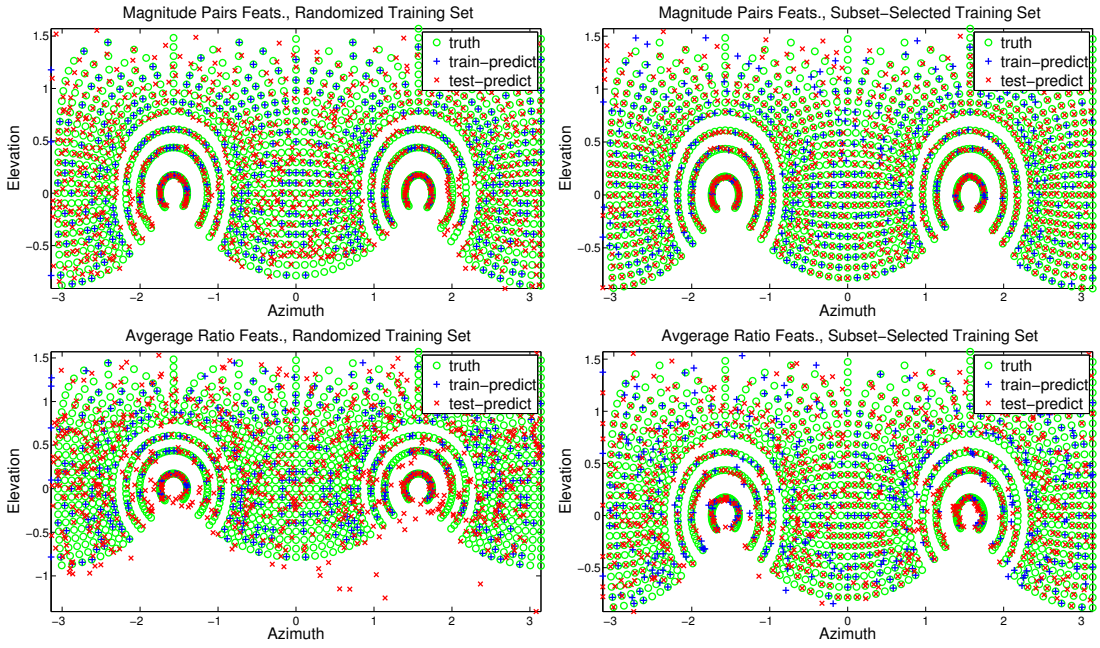


Figure 2.4: Mercator projections of GP-SSL K_{∞} predicted mean directions evidenced on randomized and subset-selected inputs (prediction error risk function R in section 2.5.2 are shown.

2.5 Feature Subset-Selection

Greedy feature selection is an efficient method for finding a subset of inputs $X_r \in X$ that best approximates a functional $f(X_r) \approx f(X)$ according to a user-specified risk function $R(X_r)$ (measure of distance between $f(X_r)$ and $f(X)$). Determining the optimal subset via an combinatorial exhaustive search is prohibitive w.r.t. the number of evaluations of R. A greedy heuristic (ranking $X_{\hat{r} \notin r}$ according to a point-inclusion in the risk evaluation $R(X_{\hat{r} \cup r})$ and adding the minimizer into the subset X_r without consideration in future iterations) reduces the search to a quadratic number of evaluations (see Algorithm 1). For GP-SSL, GFS approximates the GP posterior distribution (Eq. 2.6) evaluated on the full dataset ($X_* = X$) conditioned on a growing subset $X_{\hat{r} \cup r}$ of inputs. We propose an efficient method for updating both GP prior and posterior distributions between point-inclusions in section 2.5.1.

Algorithm 1 Greedy Forward Selection

Require: Training inputs X, y, subset size T, and risk function $R(X)$.

$r \leftarrow \emptyset$ \\ Initial empty subset at iteration $t = 0$
for $t = 1$ to T **do**
  $r \leftarrow \{r, \arg\min_{\hat{r} \notin r} R\left(X_{\hat{r} \cup r}\right)\}$ \\ Minimize risk
end for
return r

Specifying the risk function R is more difficult as its evaluation costs must be low. Most risk functions that use second-order moments (e.g. GP posterior covariance in Eq. 2.6) are expensive and require approximations to remain tractable [79]. Evaluating the GP posterior covariance requires $\mathrm{O}\left(N_*^2\right)$ space; its inverse and determinants are expensive to compute in sub-cubic time. Instead, we propose a cheaper class of risk functions that

generalizes only the first-order moments (i.e. GP posterior mean in Eq. 2.6) in section 2.5.2.

2.5.1 Incremental GP Models

A point-update to a GP model can be defined in terms of changes to the first/second moments of the GP prior and posterior distributions (Eqs. 2.5, 2.6) and both the Gram matrix $K_{(r)} = K(X_r, X_r)$ and its inverse $K_{(r)}^{-1}$ generated from inputs in X_r. While a point-update to $K_{(\hat{r} \cup r)}$ simply contains an appended row and column of covariance function evaluations $[K(X_r, x_{\hat{r}}), K(x_{\hat{r}}, x_{\hat{r}})]$, its direct inverse $K_{(\hat{r} \cup r)}^{-1}$ would be expensive to compute. Instead, we define a recurrence relation with its previous inverse $K_{(r)}^{-1}$ as follows.

Given a sample input-output pair $(\mathbf{x}_{\hat{r}}, \mathbf{y}_{\hat{r}})$ for data index $\hat{r}$, let indices $\breve{r} = r \cup \hat{r}$ be the union with the subset indices r. At iteration t, append a row and column vector along the standard basis to the Gram matrix $K_{(r)}$. The differences between $K_{(\breve{r})}$ and the appended $K_{(r)}$ are two rank-1 updates given by

$$K_{(\breve{r})} = \begin{bmatrix} K_{(r)} & k_{r\hat{r}} \\ k_{r\hat{r}}^T & k_{\hat{r}\hat{r}} \end{bmatrix} = \begin{bmatrix} K_{(r)} & 0 \\ 0 & 1 \end{bmatrix} - uu^T + vv^T, \tag{2.12}$$

$$k_{r\hat{r}} = K(X_r, X_{\hat{r}}), \quad k_{\hat{r}\hat{r}} = K(X_{\hat{r}}, X_{\hat{r}}) + \sigma^2,$$

where vectors $u = \sqrt{\frac{||w||}{2}} \left(\frac{w}{||w||} + e_t \right)$, $v = \sqrt{\frac{||w||}{2}} \left(\frac{w}{||w||} - e_t \right)$, $w = \left[-k_{r\hat{r}}^T, \frac{1 - k_{\hat{r}\hat{r}}}{2} \right]^T$, and e_t is the t^{th} column of the identity matrix. The update in Eq. 2.12 allows $K_{(\breve{r})}^{-1}$ to follow

from the modified *Woodbury* formulation [80] given by

$$
\begin{aligned}
K_{(\check{r})}^{-1} &= \bar{K}^{-1} + d_u \bar{u}\bar{u}^T - d_v \bar{v}\bar{v}^T, \quad \bar{K}^{-1} = \begin{bmatrix} K_{(r)}^{-1} & 0 \\ 0 & 1 \end{bmatrix}, \\
\bar{u} &= \bar{K}^{-1} u, \quad d_u = \left(1 - < \bar{u}, u >\right)^{-1}, \\
\bar{v} &= \left(\bar{K}^{-1} + d_u \bar{u}\bar{u}^T\right) v, \quad d_v = \left(1 + < \bar{v}, v >\right)^{-1},
\end{aligned}
\tag{2.13}
$$

which requires only two rank-1 updates. For a fixed set of test inputs X_*, the updated posterior mean vector remains a matrix-vector product and the posterior variances are sums of diagonals given by

$$
\begin{aligned}
\bar{f}_{*\check{r}} &= K_{*\check{r}} K_{(\check{r})}^{-1} Y_{\check{r}}, \quad s_u = K_{*\check{r}} \bar{u}, \quad s_v = K_{*\check{r}} \bar{v}, \\
\mathbf{diag}\left(\bar{\Sigma}_{*\check{r}}\right) &= \mathbf{diag}\left(\bar{\Sigma}_{*r} + k_{*\hat{r}} k_{*\hat{r}}^T + d_u s_u s_u^T - d_v s_v s_v^T\right),
\end{aligned}
\tag{2.14}
$$

where matrix $K_{*\check{r}} = K(X_*, X_{\check{r}})$. The updated log-determinant is given by $\log \left|K_{(\check{r})}\right| = \log \left|\bar{K}\right| - \log d_u d_v$. The total computational costs of updating the GP prior and posterior distributions at iteration t are O (t^2) and O $(N_* t)$ operations respectively.

2.5.2 GP L^2 Risk Function Criterions

We show how several risk functions can be derived from the L^2 distance between any two GP posterior mean functions evaluated at a possibly infinite sized set of test inputs X_*. Given two GPs f_a, f_b defined over the subsets of inputs X_a, X_b for indices a and b, the L^2 distance between their two GP posterior mean functions ($\bar{f}_a = K_{*a} \hat{K}_a^{-1} Y_a$ and $\bar{f}_b = K_{*b} \hat{K}_b^{-1} Y_b$) is analytic under certain GP prior assumptions. For prior mean

$m(x) = 0$ and the product of identical Matérn class covariance functions in Eq. 2.7, the errors evaluated at X_* are given by

$$\begin{aligned} L^2_{X_*}\left(\bar{f}_a, \bar{f}_b\right) &= \sum_{x_* \in X_*} (\bar{f}_a - \bar{f}_b)^2 \\ &= z_a^T Q_{aa} z_a - 2 z_a^T Q_{ab} z_b + z_b^T Q_{bb} z_b, \end{aligned} \tag{2.15}$$

where vectors $z_a = \hat{K}_a^{-1} Y_a \in \mathbb{R}^{N_a}$, $z_b = \hat{K}_b^{-1} Y_b \in \mathbb{R}^{N_b}$ are computed over training data. Updating the risk function evaluations between successive iterations t is efficient as updating $\bar{f}_a$, $\bar{f}_b$ need only rank-1 updates via Eq. 2.13. The associated matrices Q_{ab}, Q_{aa}, Q_{bb} in Eq. 2.15 are sub-matrices of Q_{XX} and can be pre-computed in O (N^2) operations. Computing Q_{XX} depends on the following cases.

Finite Case: If X_* is finite, then matrices $Q_{aa} = \sum_{x_* \in X_*} K_{a*} K_{*a} \in \mathbb{R}^{N_a \times N_a}$, $Q_{ab} = \sum_{x_* \in X_*} K_{a*} K_{*b} \in \mathbb{R}^{N_a \times N_b}$, and $Q_{bb} = \sum_{x_* \in X_*} K_{b*} K_{*b} \in \mathbb{R}^{N_b \times N_b}$ are the summation of outer-products whose i, j^{th} entries are products of Matérn class covariance functions in Eq. 2.7.

Infinite Case: If $X_* = (-\infty, \infty)$ is the full (unbounded) input domain, then matrices $Q_{aa} = \int_{-\infty}^{\infty} K_{a*} K_{*a} dx_* \in \mathbb{R}^{N_a \times N_a}$, $Q_{ab} = \int_{-\infty}^{\infty} K_{a*} K_{*b} dx_* \in \mathbb{R}^{N_a \times N_b}$, and $Q_{bb} = \int_{-\infty}^{\infty} K_{b*} K_{*b} dx_* \in \mathbb{R}^{N_b \times N_b}$ contain improper integral entries. For a valid distance measure, the posterior mean functions converge to identical zero-mean priors at the limits $x_{*k} \to \pm\infty$ and the improper integrals of the form $Q_{a_i b_j} = \prod_{k=1}^{D} F_{\nu ijk}$ given by

$$F_{\nu ijk} = \int_{-\infty}^{\infty} K_\nu(|x_{a_i k} - x_{*k}|, \ell_{ak}) K_\nu(|x_{b_j k} - x_{*k}|, \ell_{bk}) dx_{*k}, \tag{2.16}$$

are shown to be finite (see Appendix Eq. 2.23). Several combinations of the L^2 distance are summarized as follows.

Prediction Error $L^2_X\left(\bar{f}_{(\breve{r})}, y\right)$**:** The prediction error is taken between the GP posterior means $\bar{f}_{(\breve{r})}$ at test inputs $X_* = X$ and the known sample pairs (X, Y).

Generalized Error $L^2_{X_*}\left(\bar{f}_{(\breve{r})}, \bar{f}_{(X)}\right)$**:** The generalized error is taken between two GP posterior mean functions $\bar{f}_{(a)}$ and $\bar{f}_{(b)}$ evaluated at any finite X_* (may be out-of-sample from X). For GFS, the two GPs are specified by subset-selected $a = (\breve{r})$ and the full set of inputs $b = (X)$.

Normalized Error $L^2_{(-\infty,\infty)}\left(\frac{\bar{f}_{(\breve{r})}}{\left\|\bar{f}_{(\breve{r})}\right\|}, \frac{\bar{f}_{(X)}}{\left\|\bar{f}_{(X)}\right\|}\right)$**:** The normalized error or "frequentist" risk is taken between two normalized GP posterior mean functions ($\frac{\bar{f}_{(a)}}{\left\|\bar{f}_{(a)}\right\|}$ and $\frac{\bar{f}_{(b)}}{\left\|\bar{f}_{(b)}\right\|}$) evaluated at $X_* = (-\infty, \infty)$ given uniform probability distribution over x_*. The norm term $\|f\| = \sqrt{\int_{-\infty}^{\infty} f(x)^2 dx}$ is shown to be finite by setting either of the functions in Eq. 2.15 to zero. The two GPs are specified on subset-selected $a = (\breve{r})$ and the full set of inputs $b = (X)$.

2.5.3 Experiments

GFS selects for increasing subset sizes until it contains the full dataset. At each iteration t, the incremental GP-SSL K_∞ model infers directions (posterior means) along test inputs $X_* = X$. The mean angular separation error (Eq. 2.11) between the predicted and the reference measurement directions are computed and shown in Fig. 2.5; intercepts with horizontal lines indicate subset sizes at 5° and 1° errors. The crossover points at the 5° error line (localization accuracy) are achieved for MP and AMR features at a small frac-

tion of the total input set (approximately 50 and 150 feature-direction pairs); decreases in localization error after 50 randomized samples becomes logarithmic with diminishing returns. Moreover, GFS selected models generalize better than that of randomized selection in all but the PD features; a visual (second column plots in Fig. 2.4) shows that the former more accurately localizes directions further from the median plane.

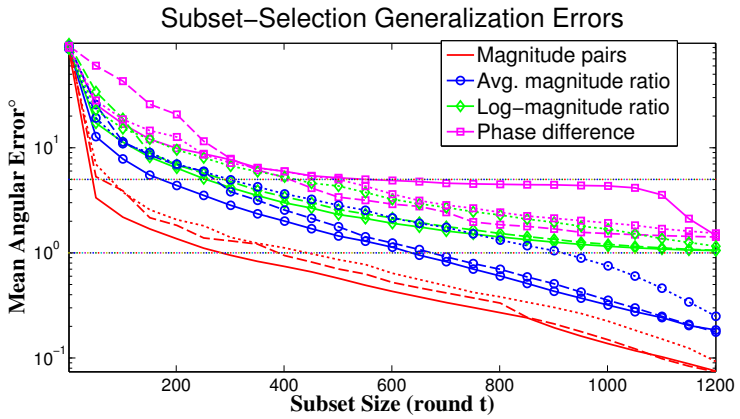


Figure 2.5: Generalization errors are shown for GP-SSL models evidenced on randomized (dotted) and GFS [prediction error (solid), normalized error (dashed)] selected subsets of feature-direction pairs.

2.6 Active-Learner System

The active-learning process for inferring HRTFs is as follows. The collection of p number of target directions is specified as $\mathbf{u} \in U \in \mathbb{R}^{3\times p}$. For rounds $t < T$, a query HRTF (MP) $\mathbf{x}_t$ is chosen from the candidate set X^C and appended to form input matrix $X \in \mathbb{R}^{T\times D}$. The listener localizes $\mathbf{x}_t$, registers the direction $\mathbf{v}_t$ over a GUI (see Fig. 2.6), and appends the directions to form matrix $V \in \mathbb{R}^{3\times T}$. The SSLEs w.r.t. U are computed in $Y_{\mathbf{u}t} = \mathbf{SSLE}(\mathbf{u}, \mathbf{v}_t)$ s.t. $Y = -U^T V \in \mathbb{R}^{p\times T}$. Last, the updated feature-direction pairs (X, Y) are added into the GP-SSLE models via incremental GPs (section 2.5.1). The system

components are organized below.

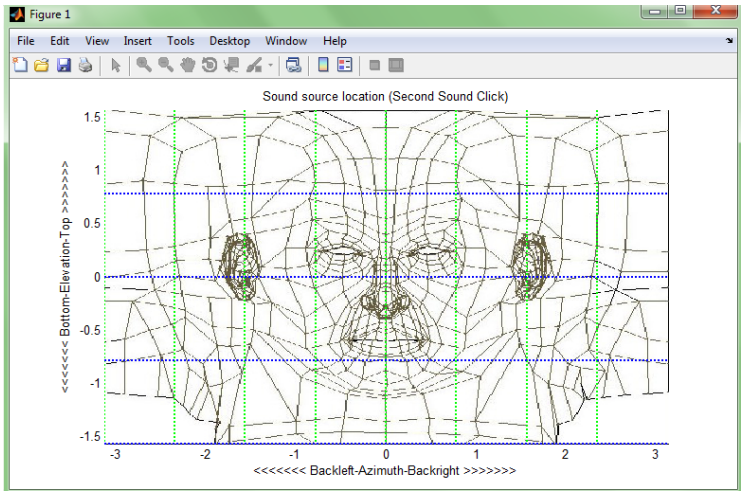


Figure 2.6: GUI shows a mercator projection of spherical coordinate system onto 2D panel. User clicks on panel to report a direction.

2.6.1 Conditional Mixture of Gaussians Models

While it is possible to specify an entire HRTF database as the candidate set, it is reasonable to assume that most samples would not be localized near a target direction **u**; overt features arising from the reflections off the anthropometry may be a physical impossibility along all measurement directions. Conversely, choosing only HRTFs with measurement directions equivalent to **u** restricts the sample size to the number of subjects in the database. To address both issues, we model both the HRTFs and their corresponding measurement directions using a conditional mixture of Gaussians model (MoG) trained from the CIPIC database (see section 2.6.1). This allows for X^C to be drawn from a distribution of HRTFs conditioned at any direction **u**.

The MoG models the joint distribution between input variables as if the samples are drawn from a latent set of normal distributions. The input variables consist of mea-

surement directions **u** and leading principal components (PCs)[7] **w** associated with HRTFs along **u**. The joint distribution is modeled by a weighted sum of M normal distributions with mean and covariances given by

$$\mathbf{z} = \begin{bmatrix} \mathbf{w} \\ \\ \mathbf{u} \end{bmatrix}, \quad \mu = \begin{bmatrix} \mu_{\mathbf{w}} \\ \\ \mu_{\mathbf{u}} \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_{\mathbf{w}} & \Sigma_{\mathbf{wu}} \\ \\ \Sigma_{\mathbf{uw}} & \Sigma_{\mathbf{u}} \end{bmatrix}, \tag{2.17}$$
$$P(\mathbf{z}) = \sum_{i=1}^{M} \pi_i \mathcal{N}\left(\mathbf{z}|\mu^{\{i\}}, \Sigma^{\{i\}}\right), \quad \sum_{i=1}^{M} \pi_i = 1,$$

where parameters μ, π, Σ are trained via the well-known expectation-maximization algorithm. The PCs **w** conditioned on **u** is also a MoG given by

$$P(\mathbf{w}|\mathbf{u}) = \sum_{i=1}^{M} \frac{\pi_i \mathcal{N}\left(\mathbf{u}|\mu_{\mathbf{u}}^{\{i\}}, \Sigma_{\mathbf{u}}^{\{i\}}\right) \mathcal{N}\left(\mathbf{w}|\mu_{\mathbf{w}|\mathbf{u}}^{\{i\}}, \Sigma_{\mathbf{w}|\mathbf{u}}^{\{i\}}\right)}{\sum_{j=1}^{M} \mathcal{N}\left(\mathbf{u}|\mu_{\mathbf{u}}^{\{j\}}, \Sigma_{\mathbf{u}}^{\{j\}}\right)}, \tag{2.18}$$

where the conditional mean and covariance for the i^{th} mixture are $\mu_{\mathbf{w}|\mathbf{u}}^{\{i\}} = \mu_{\mathbf{w}}^{\{i\}} + \Sigma_{\mathbf{wu}}^{\{i\}} {\Sigma_{\mathbf{u}}^{\{i\}}}^{-1} (\mathbf{u} - \mu_{\mathbf{u}}^{\{i\}})$ and $\Sigma_{\mathbf{w}|\mathbf{u}}^{\{i\}} = \Sigma_{\mathbf{w}}^{\{i\}} - \Sigma_{\mathbf{wu}}^{\{i\}} {\Sigma_{\mathbf{u}}^{\{i\}}}^{-1} {\Sigma_{\mathbf{wu}}^{\{i\}}}^{T}$ respectively. The candidate set X^C is given by PCs randomly sampled from the conditional MoG[8] in Eq. 2.18 and decoded into HRTFs to form the candidate set. The non-individualized (directional-averaged) HRTFs are approximated by the sum of the weighted conditional mixture means.

[7] PCs are computed from same-subject, mean-centered, log-magnitude pairs (concatenated left and right ear HRTFs).

[8] Leading 16 PCs are sampled (via Gibbs sampling) from one of $M = 64$ multivariate normal distribution (randomly selected by weight).

2.6.2 GPs for Modeling SSLE

GP-SSLE models ($f_{1:p}(X) = \{f_1(X), \dots f_p(X)\}$) are specified by a common set of input MP features X and output SSLEs Y for each of the p number target directions in U. Accurate modeling of the SSLE depends on the choice of GP prior mean and covariance functions. A zero mean prior is reasonable as reported directions $\mathbf{v}$ in the absence of localization should average to the zero vector. Choosing the GP covariance function is more difficult as the hyperparameters cannot be optimized in the absence of observations; inaccurate priors would result in poor generalizations error.

Fortunately, GP-SSLE models can be related to GP-SSL models when U is the infinite set of target directions uniformly sampled over a unit sphere. Substituting the SSLE labels $Y = -U^T V$ into Eq. 2.8, the GP-SSLE LMH is now given by

$$L = -\frac{1}{2}\left(|U|\log|\hat{K}| + \mathbf{tr}\left(QUU^T\right) + t|U|\log(2\pi)\right), \tag{2.19}$$

where matrix $Q = V\hat{K}^{-1}V^T$. As $p \to \infty$, the sample covariance of U approaches a constant variance $UU^T = \frac{1}{3}I$ due to symmetry. The LMH in Eq. 2.19 reduces to

$$L_S = -\frac{|U|}{2}\left(\log|\hat{K}| + \frac{\mathbf{tr}\,(Q)}{3} + t\log(2\pi)\right), \tag{2.20}$$

which is equivalent to that of GP-SSL models for MP features X and directions V.

The equivalence allows for the choice of the GP-SSLE model's covariance function to approximated by that of GP-SSL models trained over known feature-direction pairs

(e.g. CIPIC subject data). While these subjects are not identical to the listener, the trained GP-SSL models all share similar covariance functions as their hyperparameters are well-distributed (see Fig. 2.7); high frequency bands above 17 kHz tend to be negligible while lower frequency sub-bands between $0-3$ and $4-7.5$ kHz are relevant.

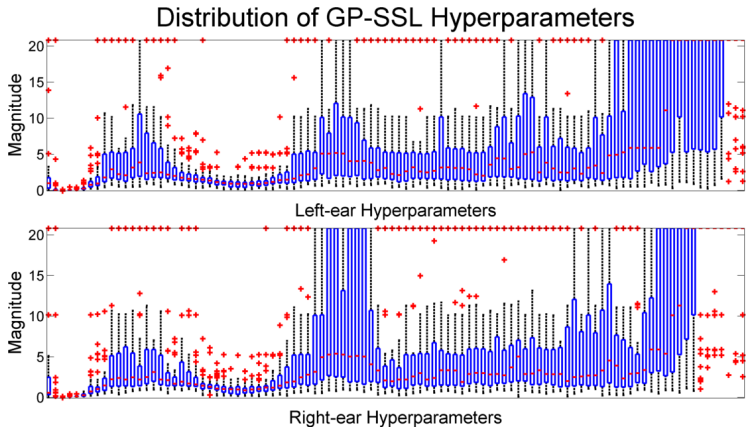


Figure 2.7: Distribution (box-plot) of hyperparameter values are shown for GP-SSL models (x-axis $0-22.1$ kHz frequency range). Large valued hyperparameters ℓ_k indicate less sensitivity along the k^{th} frequency.

2.6.3 Query-Selection

We present GP based query-selection as a modification of a known algorithm [56] which is derived as follows. Consider the observed minimum SSLE for any $\mathbf{u}$ at round t given by

$$\eta_{\mathbf{u}t} = \min(Y_{\mathbf{u}1}, \ldots, Y_{\mathbf{u}t}). \tag{2.21}$$

Realizations of SSLEs ($\gamma = f(\mathbf{x}_*|X, Y)$) by the GP-SSLE posterior distribution (Eq. 2.6) at a candidate input $\mathbf{x}_* \in X^C$ will be normally distributed whose mean and variances represent the expected SSLE and uncertainty respectively. Thus, improvements (lowering)

upon the global minimum $\eta_{\mathbf{u}t}$ is given by the loss-function $\lambda_{\mathbf{u}t}(\gamma) = \min(\gamma, \eta_{\mathbf{u}t})$ whose expectation can be computed via marginalizing over the γ.

The expected loss-function is analytic for any single $\mathbf{u}$ and so the weighted expected loss function (specified over each $\mathbf{u} \in U$ with independent GP-SSLE models) is given by

$$\begin{aligned} \wedge(x_*) &= \sum_{\mathbf{u}\in U} \rho_{\mathbf{u}} \int_{-\infty}^{\infty} \lambda_{\mathbf{u}t}(\gamma)\mathcal{N}(\gamma|\bar{\mu}_{\mathbf{u}}, \bar{C}_{\mathbf{u}})d\gamma = \sum_{\mathbf{u}\in U} \rho_{\mathbf{u}} W_{\mathbf{u}}, \\ W_{\mathbf{u}} &= \eta_{\mathbf{u}t} + (\bar{\mu}_{\mathbf{u}} - \eta_{\mathbf{u}t})\psi(\eta_{\mathbf{u}t}|\bar{\mu}_{\mathbf{u}}, \bar{C}_{\mathbf{u}}) - \bar{C}_{\mathbf{u}}\mathcal{N}(\eta_{\mathbf{u}t}|\bar{\mu}_{\mathbf{u}}, \bar{C}_{\mathbf{u}}), \end{aligned} \tag{2.22}$$

where weights $\rho_{\mathbf{u}} = 1/p$ can be set to a constant, GP-SSLE posterior mean and covariance functions at $\mathbf{x}_*$ evidenced with $(X_{1:t,:}, Y_{\mathbf{u},1:t})$ are denoted by $\bar{\mu}_{\mathbf{u}}$ and $\bar{C}_{\mathbf{u}}$, and the cumulative normal distribution of $\bar{C}_{\mathbf{u}}$ is denoted by ψ. The query HRTF is chosen as the lowest scoring candidate or minimizer $\text{argmin}_{x_*\in X^C} \wedge (x_*)$ of the criterion Eq. 2.22 which balances local improvement through the posterior mean term $(\bar{\mu}_i - \eta_t)$ with exploring uncertain predictions through the posterior variance term $\bar{C}_{\mathbf{u}}$. The property is useful for proving the rate of convergence [76] to the true solution in Eq. 2.2.

2.6.4 Experiments

GP-SSL active-learning trials: One method for fast and repeatable empirical validation substitutes the human listener for GP-SSL models trained on CIPIC subject data. Localizations at $\mathbf{x}_*$ can be reported as either the GP posterior mean directions, or by sampling from the GP posterior distribution. This allows for large subsets of X^C to be efficiently evaluated with little time costs. For coherence, we limit the query-selection criterion in Eq. 2.22 to single target directions $\mathbf{u}$ belonging to the CIPIC HRTF measurement direc-

tions (queries made for past **u** are discarded). GP-SSLE's covariance hyperparameters are set to that of the GP-SSL mean hyperparameters (averaged across 45 subject models); hyperparameters can be retrained after each round but is not necessary for improving the localization error. The variance term is set to $\sigma = 0.05$.

In tests, the active-learner submits an initial non-individualized query HRTF for **u** and then proceeds through $T = 50$ rounds of query-selection from a candidate HRTF set of 20000 samples drawn from a conditional MoG (Eq. 2.18). The nearest localized directions are shown to closer to their target directions than the non-individualized guesses (see Fig. 2.8). Non-individualized HRTFs are localized closer to the horizontal plane and towards the back of the head. Nearest localized directions accord with empirical studies of difficulties in front-back and up-down confusion with human subjects [63]. The experiment is repeated across all 45 GP-SSL CIPIC subject models (see Fig. 2.9). The improvement can be expressed as the mean ratio between the angular separation errors of the initial and nearest localized directions. The mean improvement is 7.729 across all CIPIC measurement directions, 9.139 for median plane directions, and 8.252 for horizontal plane directions.

Human active-learning trials: For a human listener, we develop a simple GUI in Matlab that consists of an azimuth-elevation plot that the subject clicks to report $\mathbf{v}_t$. To introduce contrast in hearing, two test signals are alternatively played over headphones until the listener reports a direction. The first is a short burst of WGN independently generated for left and right ear channels. The second is the WGN convolved with the left and right min-phase HRTFs derived from the binaural MP features. The trials proceed as the listener localizes queries for $T = 10$ rounds in each of the 14 target directions (7 on

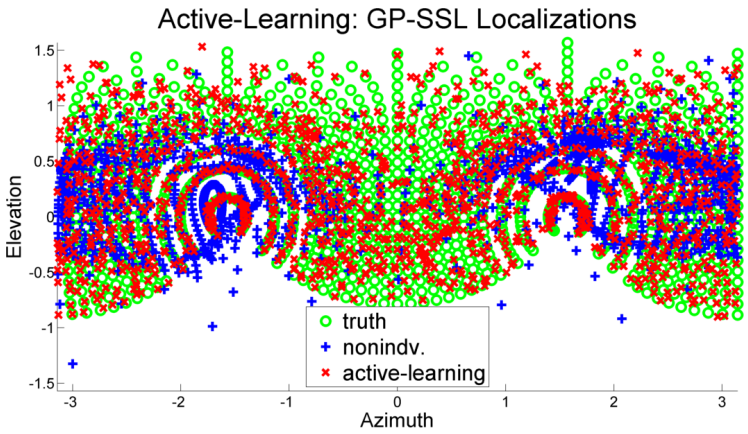


Figure 2.8: Nearest localized directions after active-learning by the GP-SSL model (red) improve upon initial non-individualized HRTF localizations (blue).

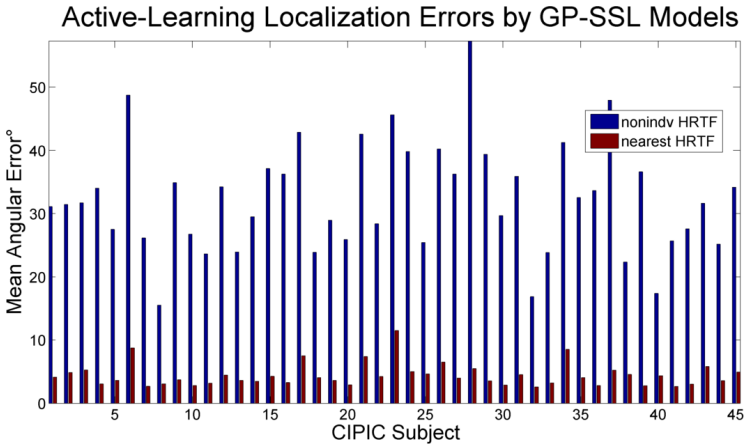


Figure 2.9: Mean angular errors are shown for the initial query (non-individualized HRTFs) and nearest HRTF queries.

the horizontal and median planes each).

For 5 sample human listeners, the initial and nearest (minimum) localization errors for each of the target direction are shown in Table 2.4 and are compared to synthetic trials conducted with the 45 GP-SSL CIPIC subject models. In both cases, the largest errors occur along the median plane direction $\theta = \{-1.6, -0.69\}$. The mean percentage improvements of the nearest localizations over that of the non-individualized HRTFs are 49% and 43% for human and GP-SSL listeners respectively. GP-SSL localization errors

are generally lower and more consistent across all direction than the human listeners; GP-SSL models can report a posterior mean direction whereas human listener exhibit variances in his/her localizations, even for identical test signals. It may be of interest in future work to both measure and model human localization variances via the GP-SSL's variance term σ and by sampling localizations from the GP posterior distribution.

Table 2.4: Active-learner: non-individualized and minimum horizontal ϕ and median θ plane localization errors (degrees)

	GP-SSL$_0$	**GP-SSL**$_{\min}$	**Human**$_0$	**Human**$_{\min}$
$\phi : -2.4$	23.1 ± 15.8	12.6 ± 9.01	42.5 ± 35.6	16.4 ± 7.43
$\phi : -1.6$	19.9 ± 12.1	10.4 ± 7.49	34 ± 14.4	5.98 ± 7.17
$\phi : -0.79$	24.6 ± 16.7	7.45 ± 4.88	56.7 ± 17.5	28.8 ± 14
$\phi : 0.79$	22 ± 16.2	7.87 ± 5.12	48.7 ± 18	21.5 ± 13.6
$\phi : 1.6$	15.8 ± 9.38	6.63 ± 3.68	23.7 ± 10.6	10.8 ± 5.23
$\phi : 2.4$	22.7 ± 14.7	13.2 ± 7.06	31.2 ± 11.6	14.9 ± 5.26
$\theta : -1.6$	55.6 ± 26	37.1 ± 20.8	119 ± 43.3	59.8 ± 29.5
$\theta : -0.79$	105 ± 44.9	37.9 ± 20.9	104 ± 37.3	61.8 ± 22.4
$\theta : 0$	44.1 ± 44	11.6 ± 9.75	39.2 ± 22.1	23.3 ± 9.82
$\theta : 0.79$	35.9 ± 23.2	15.8 ± 11.1	24.7 ± 12.3	15.3 ± 4.76
$\theta : 1.6$	31.9 ± 18.4	15.6 ± 9.5	55 ± 23.1	30.2 ± 25.9
$\theta : 2.4$	17.2 ± 14.8	10.8 ± 7.38	83.6 ± 56	24.3 ± 23.9
$\theta : 3.1$	24.5 ± 19.6	12.6 ± 6.88	92.7 ± 68.1	11.9 ± 8.72
$\theta : 3.9$	26.1 ± 17.1	8 ± 5.67	61.5 ± 42.7	18.6 ± 11.1

2.7 Conclusions

We developed a robust method for the SSL using sound-source invariant features derived from left and right ear HRTF measurements. Our GP-SSL models generalized NN based approaches and were shown to more accurate in both cases of randomized and subset-selected features; good spatialization accuracy (5°) over the full sphere was possible using a fraction of the available features. For learning HRTFs in listening tests, we developed an active-learning method for query-selection using GP models. Both simulations with offline GP-SSL models and HRTFs recommended to real human listeners have shown

large improvement in localization accuracy over non-individualized HRTFs.

2.8 Appendix: Matérn Product Integrals

Improper integrals in Eq. 2.16 have closed-formulations:

$$
\begin{aligned}
F_{\frac{1}{2}ijk} &= \left(\ell_{ak} e^{\frac{-\left|x_{a_i k} - x_{b_j k}\right|}{\ell_{ak}}} - \ell_{bk} e^{\frac{-\left|x_{a_i k} - x_{b_j k}\right|}{\ell_{bk}}} \right) \frac{2\ell_{ak}\ell_{bk}}{\ell_{ak}^2 + \ell_{bk}^2}, \\
F_{\frac{3}{2}ijk} &= \left(\ell_{ak}^2(\ell_{ak} - \beta\ell_{bk} - \alpha) e^{\frac{-\sqrt{3}\left|x_{a_i k} - x_{b_j k}\right|}{\ell_{ak}}} \right. \\
&\quad \left. + \ell_{bk}^2(\ell_{bk} + \beta\ell_{ak} - \alpha) e^{\frac{-\sqrt{3}\left|x_{a_i k} - x_{b_j k}\right|}{\ell_{bk}}} \right) \frac{4\ell_{ak}\ell_{bk}}{\sqrt{3}\left(\ell_{ak}^2 - \ell_{bk}^2\right)^2}, \\
\alpha &= -\sqrt{3}\left|x_{a_i k} - x_{b_j k}\right|, \quad \beta = \frac{4\ell_{ak}\ell_{bk}}{\ell_{ak}^2 - \ell_{bk}^2}, \\
F_{\infty ijk} &= e^{-\frac{\left(x_{a_i k} - x_{b_j k}\right)^2}{2\left(\ell_{ak}^2 + \ell_{bk}^2\right)}} \frac{\ell_{ak}\ell_{bk}\sqrt{2\pi}}{\sqrt{\ell_{ak}^2 + \ell_{bk}^2}}.
\end{aligned}
\tag{2.23}
$$

Chapter 3: Fast Sparse and Gridded Gaussian Process Regression for HRTF Interpolation

3.1 Introduction

Bayesian non-parametric kernel methods such as Gaussian processes (GPs) [46] have successfully been used for many regression and classification problems. However, the high computational costs of GP regression (GPR) presents a major bottleneck for learning on large datasets. For N data points, naive GPR requires $\mathrm{O}\left(N^3\right)$ operations and $\mathrm{O}\left(N^2\right)$ space due to inverting a large covariance (Gram) matrix or the equivalent of solving for a large matrix system. This is problematic for both real-time inference and off-line training phases where these operations must be repeated, often for an unspecified number of test data and a large number of training samples.

Fortunately for some datasets (X, y) (input features, output observations), it is possible to design GP models to take advantage of structures inherent to its data organization. For *gridded* datasets [81, 82], large computational savings are possible under the following conditions: First, observations y are parameterized by D-dimensional inputs $x \in X$ where $X = X_1 \times X_2 \times \ldots \times X_D$ is the set of D-Cartesian outer products between subsets $X_i \in \mathbb{R}^{m_i \times *}$ of m_i elements each. Second, the GP covariance function has the form

of a tensor product kernel (TPK) $K(x_j, x_k) = \prod_{i=1}^{D} K_i(x_j^{[i]}, x_k^{[i]})$ where function K_i is restricted to inputs belonging to the input subset X_i. These two conditions allow for the covariance matrix to be expressed as Kronecker tensor products (KTPs) [57] given by $C = \otimes_{i=1}^{D} C_i$ and D-Kronecker factors $C_i \in \mathbb{R}^{m_i \times m_i}$ for $N = \prod_{i=1}^{D} m_i$ number of inputs. The Kronecker product $\otimes$ is a binary operation between matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{p \times q}$ that generates the block matrix given by

$$A \otimes B = \begin{pmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn}B \end{pmatrix} \in \mathbb{R}^{mp \times nq}. \tag{3.1}$$

See Appendix 3.9.1 for a list of KTP identities.

Prior works have extended these TPK structured covariances for GP inference and training, especially under noisy conditions (addition of a noise term to the model) [83]. The noise term manifests as diagonalized entries added to the covariance matrix which violates the conditions for direct KTP decomposition; several works have proposed various treatments of this problem: For variable noise, an independent GP can be trained to separately model the noise terms over separate inputs [84]. For constant white Gaussian noise (WGN), low-rank approximations of the covariance matrix for low dimensions ($D = 2$) can be computed [85]. The general case of isotropic WGN can be handled via the "eigendecomposition trick" (GPR_GRID algorithm [81]). Later works use this technique for efficient GP inference and hyperparameter training [86], [87]. We refer to these algorithms collectively as "grid GPR" and introduce a number of extensions outlined be-

low:

In Section 3.2.1, we establish notation and several matrix algorithms that use the Kronecker structure. The formulation for grid GPR is derived in section 3.2.2 where the costs of grid GPR inference and hyperparameter training (gradient evaluations) are greatly reduced. In section 3.2.3, we remark on a connection between TPK covariances and multidimensional grids to Gaussian process latent variable models (GPLVM) [88]. GPLVM is a method for dimensionality reduction that maps a lower dimensional latent space to the original data constrained by a prior covariance function; learning the latent inputs can be done by optimizing w.r.t. the data log-marginal likelihood (LMH) function. We show that GPLVM's LMH has a Kronecker product formulation where the unconstrained latent inputs are mapped to a $D = 2$ grid. Generalizations to higher dimensions require the additional constraint that the latent points also lie on a multidimensional grid.

In Section 3.3, we extend the TPK and multidimensional grid conditions to sparse GPR methods [58]. Sparse GPR makes a tradeoff between computational costs and accuracy by making an additional assumption on the conditional independence of joint prior random function evaluations given a small set of M *inducing variables*. Works such as GPML [77] and SPGP/GPSTUFF [89] show that a GP spanned by a small number ($M << N$) of these inducing inputs can often approximate the full GP at reduced costs (inference and hyperparameter training require $\mathrm{O}\left(M^2N\right)$ operations). We show that greater computational gains are possible when the multidimensional grid and TPK conditions are extended to both training and inducing inputs. Moreover, we address the case where the sparsity assumption does not hold for all dimensions in the inducing inputs and show that some forms of the *economical* Gram matrix [58] have efficient KTP

formulations.

In Sections 3.4 and 3.5, we present efficient methods for handling cases of missing data inputs (holes in the multidimensional grid) and extra data (points outside a grid) respectively. Since both cases marginally violate the gridded conditions, we show how GP inference and hyperparameter training can remain both efficient and exact as if performed using grid GPR with reduced costs. Furthermore, this relaxation generalizes standard GPR with TPK covariances as any input set can be contained or appended from a multidimensional grid that spans the Cartesian outer products between all unique inputs along each input dimension.

In Section 3.6, we extend the method of greedy backward subset-selection (GBSS) [54] to grid GPR for "ranking" input samples according to GP prior assumptions. GBSS begins with a full set of inputs and iteratively eliminates input samples that minimizes/maximizes a specified objective function; no input is considered twice after it has been eliminated. For ranking, we are interested in finding the largest subset of samples that satisfy the GP prior assumptions; we thus specify grid GP's remaining data (samples not eliminated) LMH as the objective function. Moreover, the formulation naturally extends our previous result on the efficient handling of missing data in grid GPs as the eliminated samples are equivalent to the missing data subset.

Last in Section 3.7, we perform a large array of experiments on both synthetic data and real head-related impulse response/transfer functions (HRIR/HRTF)[1] datasets[2]. Section 3.7.1 demonstrates runtime gains on synthetic data for cases of variable input dimen-

[1]Magnitude HRTF responses can be mapped from a 2D multidimensional grid formed by a tensor product between spherical coordinates and frequency domain inputs

[2]CIPIC database [1]

sions and missing/extra inputs. Section 3.7.2 demonstrates runtime-to-accuracy trade-offs for HRTF interpolation via grid and sparse-grid GPR methods compared to other spherical interpolation methods from literature. Section 3.7.2.3 shows how further computational gains can be achieved via series expansion methods demonstrated on a gridded spherical domain. Section 3.7.2.4 shows how local features such as the HRTF spectral extrema can be extracted. Section 3.7.3 applies subset-selection methods to the problem of interpolating HRTF interaural time differences (ITDs) in the spherical coordinate domain. Section 3.7.4 applies subset-selection methods to extracting a parsimonious set of inputs for perceptually relevant reconstructions of missing HRTF measurements.

3.2 GPR Background

Formally, a GP is a collection of random variables $\mathbf{f} = [f(x_1), f(x_2), \ldots, f(x_N)]$ indexed at $X = [x_1, x_2, \ldots, x_N]$ such that any finite subset is jointly Gaussian; realizations of $\mathbf{f}$ generate a vector of random function values drawn from a $N-$variate Gaussian distribution. The distribution is specified by the GP prior mean $m(x) = 0$ (without loss of generality) and covariance (cov) function $K(x_i, x_j)$ between the function evaluations $f(x_i), f(x_j)$ in the form of

$$f(x) \sim GP(m(x), K(x_i, x_j)), \quad K(x_i, x_j) = \mathbf{cov}\left(f(x_i), f(x_j)\right). \tag{3.2}$$

For the general regression problem, observations y are generated (realized) from a latent function $f(x)$ (treated as a random variable indexed on variables x), and corrupted by

Gaussian white noise:

$$y = f(x) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2), \tag{3.3}$$

where the noise term ϵ is zero centered with constant variance σ^2. For GP $f(x)$ with prior zero mean and covariance function K, the joint distribution between training outputs $f(x) + \epsilon = y$ (at known input x) and the test output $f_* = f(x_*)$ (at input x_*) is given by

$$\begin{aligned} \begin{bmatrix} y \\ f_* \end{bmatrix} &\sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) + \sigma^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right), \\ K_{ff} &= K(X,X), \quad \hat{K} = K_{ff} + \sigma^2 I, \\ K_{f*} &= K(X, X_*), \quad K_{**} = K(X_*, X_*), \end{aligned} \tag{3.4}$$

where X and X_* are the collection of training and test inputs respectively. For inference, the test output $f(X_*)$ conditioned on $f(X) = y$ (test input, training data, and training inputs) is normally distributed $P(f_*|X, y, X_*) \sim \mathcal{N}(\bar{f}_*, \mathbf{cov}\,(f_*))$ with a predicted mean (expectation) and predicted covariance (uncertainty) given by

$$\bar{f}_* = E[f_*|X, y, X_*] = K_{f*}^T \hat{K}^{-1} y, \quad \mathbf{cov}\,(f_*) = K_{**} - K_{f*}^T \hat{K}^{-1} K_{f*}. \tag{3.5}$$

Thus, inference produces a *posterior mean* and *posterior covariance* at the test output f_* which are fully specified by the covariance function K and training outputs y in Eq. 3.5 via the representer theorem.

The choice of the prior covariance function K determines the "smoothness/correlatedness"

of latent function realizations $f(X)$ at nearby X. For example, the infinitely differentiable covariance function (w.r.t. x) such as the squared exponential $k(x_i, x_j) = \exp\left(-\frac{(x_i - x_j)^2}{2\theta^2}\right)$ will generate very smooth functions $f(X)$. The goodness-of-fit of observations y w.r.t. the GP prior assumptions can be evaluated by marginalizing the data likelihoods (y from $f(x) + \epsilon$) and priors (realizations of $f(x)$ drawn from the GP prior distribution) over all possible realizations of $f(x)$; this quantity is the so-called "data log-marginal likelihood" (LMH) and obtains an analytic form that is useful for evaluating the selection of covariance functions. Moreover, covariance functions can be further characterized by their hyperparameters (Θ_i) that describe their various qualities such as the function's weight, rate-of-decay to zero (eigenfunctions), and periodicity. As continuous values, hyperparameters can be optimized by maximizing the data LMH via hill-climbing methods such as steepest ascent. Both the LMH and its partial derivative w.r.t. Θ_i are given by

$$\begin{aligned} \log p(y|X) &= -\frac{1}{2}\left(\log|\hat{K}| + y^T\hat{K}^{-1}y + N\log(2\pi)\right), \\ \frac{\partial \log p(y|X)}{\partial \Theta_i} &= -\frac{1}{2}\left(\mathbf{tr}\left(\hat{K}^{-1}P\right) - y^T\hat{K}^{-1}P\hat{K}^{-1}y\right), \end{aligned} \tag{3.6}$$

where matrix $P = \partial\hat{K}/\partial\Theta_i$.

The overall computational complexity of GPs can thus be summarized by both the cost of finding informative priors, namely model-order selection via covariance function hyperparameter optimization (Eq. 3.6), and the cost of GP inference (Eq. 3.5). The relevant linear algebra operations include the matrix inversion of $\hat{K}$ (solving the matrix system $t = \hat{K}^{-1}y$, t^TPt), computing the matrix determinant $\log|\hat{K}|$, and computing the trace term $\mathbf{tr}\left(\hat{K}^{-1}P\right)$. The asymptotic costs are thus $\mathrm{O}(N^3)$ operations and $\mathrm{O}(N^2)$

space.

3.2.1 Kronecker Product Methods for GPR

Many linear algebra operations for Kronecker product matrices can be efficiently computed as a result of its block structure (see appendix 3.9.1 for a list of properties). For notation, we refer to covariance matrices K_{f*}, K_{ff}, $\hat{K}$ from Eq. 3.4 as C_{f*}, C, and $\hat{C}$ respectively as to not be confused with the selection and evaluation of the covariance function. Let the covariance matrix C and its partial derivative matrix P w.r.t. parameter $\Theta_\pi \in K_j$ have the D-KTP decompositions given by

$$C = C_1 \otimes C_2 \cdots \otimes C_D = \otimes_{i=1}^{D} C_i, \quad P = \otimes_{i=1}^{j-1} C_i \otimes \frac{\partial C_j}{\partial \Theta_\pi} \otimes_{l=j+1}^{D} C_l, \tag{3.7}$$

where $\otimes$ follows from Eq. 3.1. Efficient methods for the eigendecomposition of a D-KTP matrix, vector-Kronecker tensor product (VKTP) and Kronecker tensor-vector product (KTVP) are presented as follows.

Eigendecomposition: The eigendecomposition of each real symmetric positive-definite factor matrix is given by $C_i = U_i Z_i U_i^T$, where matrices U_i and Z_i, are the eigenvectors and the diagonal matrix of eigenvalues respectively. Let the partial derivative factor matrix P have an analogous eigendecomposition given by $\partial C_j / \partial \Theta_\pi = V_j W_j V_j^T$. This decomposition allows the full matrix inverse C^{-1} and partial derivative matrix P in Eq. 3.7 to be expressed as matrix products of KTP eigenvectors U and diagonalized

eigenvalues Z (diagonal scaling) given by

$$\begin{aligned} &U = \otimes_{i=1}^{D} U_i, \quad Z = \otimes_{i=1}^{D} Z_i, \quad C^{-1} = UZ^{-1}U^T, \\ &V = \otimes_{i=1}^{j-1} U_i \otimes V_j \otimes_{l=j+1}^{D} U_l, \quad W = \otimes_{i=1}^{j-1} Z_i \otimes W_j \otimes_{l=j+1}^{D} Z_l, \quad P = VWV^T. \end{aligned} \tag{3.8}$$

Thus, the costs of the eigendecomposition of C are summed over the eigendecompositions of individual Kronecker factor matrices C_i which require $\mathrm{O}\left(\sum_{i=1}^{D} m_i^3\right)$ operations and $\mathrm{O}\left(\sum_{i=1}^{D} m_i^2\right)$ space. Both quantities are smaller than the costs of standard naive GPR ($\mathrm{O}\left(N^3\right)$ operations and $\mathrm{O}\left(N^2\right)$ space where $N = \prod_{i=1}^{D} m_i$).

VKPT: A VKTP is the Kronecker product between D number of column-vectors (each column belongs to a Kronecker factor) specified by $D-$dimensional KTP column-indices $\bar{q} \in \mathbb{N}^D$; column index $\bar{q}_i | 1 \leq \bar{q}_i \leq m_i$ refers to the $\bar{q}_i^{th}$ element in set X_i. The notation may be used to parameterize (index) a scalar observation in y, or interchanged with general column index notation $q \in \mathbb{N}$ where $q | 1 \leq q \leq N$ and $x_q = \{x_{\bar{q}_1}^{[1]}, \ldots, x_{\bar{q}_D}^{[D]}\}$; scalar index q parameterizes the q^{th} observation y_q (see Algorithms 7 and 8 in appendix 3.9.5 for converting between q and $\bar{q}$). The VKTP and the diagonal of a KTP are given by

$$\mathbf{VKTP}\left(X, x_q \in X\right) = \otimes_{i=1}^{D} K_i(X_i, x_{\bar{q}_i}^{[i]}), \quad \mathbf{diag}\left(C\right) = \otimes_{i=1}^{D} \mathbf{diag}\left(C_i\right), \tag{3.9}$$

for the covariance evaluations $K_i(X_i, x_{\bar{q}_i}^{[i]})$ between all elements in X_i and element $x_{q_i}^{[i]}$ which require $\mathrm{O}\left(N\right)$ operations and space.

KTVP: The generalized KTVP is the matrix-vector product between a rectangular D-KTP matrix $C = \otimes_{i=1}^{D} C_i \in \mathbb{R}^{m_i \times \ddot{m}_i}$ and the vector $y \in \mathbb{R}^{\ddot{N}}$ where $\ddot{N} = \prod_{i=1}^{D} \ddot{m}_i$. The formulation is derived by first decomposing matrix C into D-matrix products of three

KTPs given by

$$M_i = \prod_{j=1}^{i-1} m_j, \quad \ddot{M}_i = \prod_{j=i+1}^{D} \ddot{m}_j, \quad C = \otimes_{i=1}^{D} C_i = \prod_{i=1}^{D} I_{M_i} \otimes C_i \otimes I_{\ddot{M}_i}, \tag{3.10}$$

and expressing vector y by its vectorization $y = \mathbf{vec}(Y)$ (stacking the columns of a matrix Y). From Eq. 3.10, a single vectorized matrix-vector product is given by

$$\left((I_{M_i} \otimes C_i) \otimes I_{\ddot{M}_i}\right) y = \mathbf{vec}(Y(I_{M_i} \otimes C_i^T)), \tag{3.11}$$

for matrix $Y \in \mathbb{R}^{\ddot{M}_i \times \ddot{m}_i M_i}$ and block-diagonal matrix $I_{M_i} \otimes C_i^T$. For standard GPR where the covariance matrices C_i are square ($m_i = |X_i|$, $N = \prod_{i=1}^{D} m_i$ and $m_i = \ddot{m}_i$), computing the block diagonal matrix-vector product in Eq. 3.11 requires $\mathrm{O}\left(m_i N\right)$ operations making the total cost of a KTVP $\mathrm{O}\left(N \sum_{i=1}^{D} m_i\right)$ operations and $\mathrm{O}\left(N + \sum_{i=1}^{D} m_i^2\right)$ space. For rectangular covariance matrix $C \in \mathbb{R}^{N \times \ddot{N}}$, the KTVP equivalent method [81] is easily modified by updating the number of entries after each matrix-vector product in Algorithm 2 where the ratio between covariance sizes $\rho_j = m_j / \ddot{m}_j$ factors into the total cost given by $\mathrm{O}\left(\ddot{N} \sum_{i=1}^{D} m_i \prod_{j=i+1}^{D} \rho_j\right)$ operations. This is used in our extension of grid conditions to sparse-grid GPR methods.

3.2.2 Grid GPR and Cost Analysis

The multidimensional gridded inputs and TPK assumptions for GP covariance matrices to be expressed as KTPs, which result in significant savings. This is ideal in the noiseless case ($\sigma = 0$) as the subsequent computations are straight-forward: Computing

Algorithm 2 Generalized Kronecker tensor-vector product (**KTVP**)

Require: Kronecker factors $[C_1 \in \mathbb{R}^{m_1 \times \ddot{m}_1}, \dots C_D \in \mathbb{R}^{m_D \times \ddot{m}_D}]$, vector $y \in \mathbb{R}^{\ddot{N}}$
1: $n \leftarrow \ddot{N} = \prod_{i=1}^{D} \ddot{m}_i$
2: **for** $i = D$ to 1 **do**
3: $\quad Y \leftarrow \text{reshape}(y, \ddot{m}_i, n/\ddot{m}_i)$
4: $\quad Y \leftarrow C_i Y \quad \backslash\backslash$ Matrix-matrix product after vectorizing y in Eq. 3.11
5: $\quad Y \leftarrow Y^T$
6: $\quad y \leftarrow \textbf{vec}(Y)$
7: $\quad n \leftarrow \text{length}(y) \quad \backslash\backslash$ Update length of vector y
8: **end for**
9: **return** $y \in \mathbb{R}^{N=\prod_{i=1}^{D} m_i}$

matrix $\hat{C} = C$ follows from the KTP decomposition, matrix-vector terms $t = C^{-1}y$ and $t^T P t$ from successive KTVPs (Eq. 3.8), log-determinant of matrix $\hat{C}$ from the log-sum of the eigenvalues specified by matrix Z, and the trace of the matrix product $\hat{C}^{-1}P = \otimes_{i=1}^{D} C_i^{-1} P_i$ from the sum of its diagonal entries (Eq. 3.9).

For non-zero isotropic noise ($\sigma > 0$), we rely on eigendecomposition of covariance matrix C into products of KTP eigenvectors for all Kronecker factors C_i; incorporating the noise term is simply the addition of the bandwidth term σ^2 to the diagonal of the KTP eigenvalue matrix Z. The subsequent operations for log-determinant and inverse-covariance matrix-vector product are given by

$$\log|\hat{C}| = \sum_{i=1}^{N} \log\left(\textbf{diag}\left(Z\right)_i + \sigma^2\right), \quad \hat{C}^{-1} = (C + \sigma^2 I)^{-1} = U(Z + \sigma^2 I)^{-1} U^T,$$
$$t = \hat{C}^{-1} y = \textbf{KTVP}\left(U, [1./\textbf{diag}\left(Z + \sigma^2 I\right)]. * \textbf{KTVP}\left(U^T, y\right)\right), \tag{3.12}$$

using two KTVPs and diagonal scaling operations; computing the term $t^T P t$ follows a similar procedure for partial derivative KTP matrix P. Computing the trace term, using

the invariance under cyclic permutation property and the inner product of diagonals from Eq. 3.9, is given by

$$\mathbf{tr}\left(\hat{C}^{-1}P\right) = \mathbf{diag}\left((Z+\sigma^2 I)^{-1}\right)^T \mathbf{diag}\left(\otimes_{i=1}^{j-1} Z_i \otimes U_j^T V_j W_j V_j^T U_j \otimes_{l=j+1}^{D} Z_l\right), \tag{3.13}$$

where the costs of all diagonalizations in Eq. 3.13 is O $\left(N + m_j^3\right)$ operations.

The total costs of grid GPR inference (mean prediction) and computing the LMH gradient are reduced to O $\left(\sum_{i=1}^{D} m_i^3 + N\sum_{i=1}^{D} m_i\right)$ operations and O $\left(N + \sum_{i=1}^{D} m_i^2\right)$ space respectively. Both terms are minimized when the size of each Kronecker factor C_i is the constant $m_i = N^{1/D}$ with best case sub-quadratic asymptotic costs of O $\left(D(N^{3/D} + N^{1+1/D})\right)$ operations and O $\left(N + DN^{2/D}\right)$ space.

3.2.3 Relation to GPLVM

GPLVM [88] is a technique for dimensionality reduction that maps a set of $\tilde{d}_l$-dimensional latent variables $X \in \mathbb{R}^{\tilde{N}\times\tilde{d}_l}$ to the set of $\tilde{d}$-dimensional observations $Y \in \mathbb{R}^{\tilde{N}\times\tilde{d}}$ that is constrained by a covariance prior. Finding the unconstrained latent variables X (low-dimensional representation) can be achieved by optimizing the GP data goodness-of-fit criterion, namely the LMH function. We show that the LMH formulation is simply the inverse formulation of grid GPR by expressing trace and log-determinant terms as

$$\begin{aligned} \log p(Y|X) &= -\frac{1}{2}\left(\tilde{d}\log|\hat{C}| + \mathbf{tr}\left(Y^T\hat{C}^{-1}Y\right) + N\log(2\pi)\right), \\ \mathbf{tr}\left(Y^T\hat{C}^{-1}Y\right) &= y^T\tilde{C}^{-1}y, \quad \tilde{C} = I_{\tilde{d}}\otimes C + \sigma^2 I_N, \quad \tilde{d}\log|\hat{C}| = \log|\tilde{C}|, \end{aligned} \tag{3.14}$$

where vector $y = \mathbf{vec}(Y) \in \mathbb{R}^N$, and $N = \tilde{N}\tilde{d}$ (see Appendix 3.9.2).

Grid GPR's formulation (Eq. 3.14) from that of GPLVM, is interpreted as the addition of latent inputs $X_1 = [1, \ldots, \tilde{d}]^T$ and a leading Kronecker delta covariance function. When latent variables X are not constrained to a multidimensional grid, GPLVM becomes grid GPR for $D = 2$ and mean prediction and gradient computations have compact forms. The log-determinant and inverse matrix product $t = \tilde{C}^{-1}y$ can be expressed as a discrete time Lyapunov or Sylvester equation [90] given by

$$\log|\tilde{C}| = \tilde{d}\sum_{i=1}^{\tilde{N}} \log\left(\mathbf{diag}\left(Z\right)_i + \sigma^2\right), \quad C^T T + \sigma^2 T = Y, \quad t = \mathbf{vec}(T),$$

which has a standard solution [91]. The gradient terms for partial derivative matrices $\tilde{P} = I_{\tilde{d}} \otimes P$ and $P = \partial C/\partial\Theta_i$ are expressed as

$$t^T\tilde{P}t = t^T\mathbf{vec}(P^T T), \quad \mathbf{tr}\left(\tilde{C}^{-1}\tilde{P}\right) = \mathbf{diag}\left(\left(I_{\tilde{d}} \otimes Z + \sigma^2 I_N\right)^{-1}\right)^T \mathbf{diag}\left(I_{\tilde{d}} \otimes (U^T P U)\right).$$

When the latent inputs are also be constrained to a multidimensional grid, the covariance matrix decomposes into KTPs in addition to the leading identity-block matrix.

3.3 Sparse-Grid GPR

Sparse GPR methods are commonly used for large datasets that reduce the $\mathrm{O}\left(N^3\right)$ computational overhead of standard GPR to a more manageable $\mathrm{O}\left(M^2 N\right)$ for $M << N$ number of sparse or "inducing" variables which summarize latent function realizations along both training and test inputs. For tensor datasets, the sparsity assumption may not

apply to each dimension of the grid; the number of inducing inputs M would be the product of sparse and dense dimension sizes which may be large. We show below that some sparse GPR methods have efficient formulations for inference and hyperparameter training for gridded inducing variables.

For notation, a unified framework for sparse GPR [58] was presented as a modification of the joint prior $p(f, f_*)$ under the additional assumption that all latent function realizations $\mathbf{f}$ are conditionally independent given a set of M random (inducing) variables $\mathbf{u} = [u_1, \ldots, u_M]^T$ indexed by the input set $X^{\{u\}}$. The approximated joint priors $q(y, f_*)$, after marginalizing out the inducing variables $\mathbf{u}$, has the normal distribution given by

$$q(y, f_*) \sim \mathcal{N}\left(0, \begin{bmatrix} \hat{Q} & Q_{f*} \\ Q_{*f} & c \end{bmatrix}\right), \quad \hat{Q} = Q_{ff} + \wedge, \quad Q_{ff} = K_{fu} K_{uu}^{-1} K_{uf},$$

for the matrices $K_{uu} = K(X^{\{u\}}, X^{\{u\}})$, $K_{fu} = K(X, X^{\{u\}})$ and the terms $(\wedge, c)$ which depend on the sparse method. The modified LMH function and its gradient w.r.t. hyperparameter Θ_i have the same formulation as Eq. 3.6 with substitution $\hat{Q} \to \hat{K}$. This allows hyperparameters and inducing inputs $X^{\{u\}}$ (treated as hyperparameters) to be trained using gradient methods. Both the posterior means and posterior variances have the respective expanded and compact formulations given by

$$\begin{aligned} q(f_*|y) &= \mathcal{N}(Q_{*f}(Q_{ff} + \wedge)^{-1} y, c - Q_{*f}(Q_{ff} + \wedge)^{-1} Q_{f*}) \\ &= \mathcal{N}(K_{*u} \Sigma K_{uf} \wedge^{-1} y, c - Q_{**} + K_{*u} \Sigma K_{u*}), \quad \Sigma = (K_{uf} \wedge^{-1} K_{fu} + K_{uu})^{-1}. \end{aligned} \tag{3.15}$$

The latter formulation requires the inversion of a so-called "economical" Gram matrix $\Sigma \in \mathbb{R}^{M \times M}$ which requires $\mathrm{O}\left(M^2 N\right)$ operations and $\mathrm{O}\left(M^2\right)$ space to compute.

To extend our grid GPR conditions for sparse GPR, we first consider both subset of regressors (SoR) and deterministic training conditional (DTC) sparse methods where $\wedge_{\text{SoR}} = \wedge_{\text{DTC}} = \sigma^2 I$, $c_{\text{SoR}} = Q_{**}$, $c_{\text{DTC}} = K_{**}$. In these case, only the matrix terms K_{uu}, $K_{uf}K_{fu}$ and $K_{uf}y$ are relevant for analysis.

Case 1: For multidimensional gridded inputs X and arbitrary inducing inputs $X^{\{u\}}$, the rows of matrix K_{uf} are expressed as $\mathbf{VKTP}\left(X, x_u \in X^{\{u\}}\right)^T$ using Eq. 3.9. The matrix products $K_{uf}y$ and $K_{uf}K_{fu}$ require $\mathrm{O}\left(MN\right)$ and $\mathrm{O}\left(M^2 \sum_{i=1}^{D} m_i\right)$ operations with $\mathrm{O}\left(M\right)$ and $\mathrm{O}\left(M^2\right)$ space respectively.

Case 2: For arbitrary inputs X and multidimensional gridded inducing inputs $X^{\{u\}} = X_1^{\{u\}} \times X_2^{\{u\}} \times \ldots \times X_D^{\{u\}}$ ($m_i^{\{u\}} = |X_i^{\{u\}}|$, $M = \prod_{i=1}^{D} m_i^{\{u\}}$), the matrix products $K_{uf}y$ and $K_{uf}K_{fu}$ are outer-products $\sum_{i=1}^{N} K(X^{\{u\}}, x_i) y_i$ and $\sum_{i=1}^{N} K(X^{\{u\}}, x_i) K(x_i, X^{\{u\}})$, computable in $\mathrm{O}\left(MN\right)$ and $\mathrm{O}\left(M^2 N\right)$ operations with $\mathrm{O}\left(M\right)$ and $\mathrm{O}\left(M^2\right)$ space respectively.

Case 3: For multidimensional gridded inputs X and inducing inputs $X^{\{u\}}$, the matrices K_{fu}, K_{uf}, and K_{uu} have both a KTP factorization and the low-rank decomposition following the example of $K_{uf}K_{fu} = \otimes_{i=1}^{D} K_i^{\{uf\}} K_i^{\{fu\}}$. If matrix Σ is stored, then matrix operations $K_{uf}y$ and $K_{uf}K_{fu}$ depend on a sparsity ratio (number of inputs along dimension over total number of inputs) given by $\rho_j = m_j^{\{u\}}/m_j$; computing these matrices

require

$$\mathrm{O}\left(N\sum_{i=1}^{D} m_i^{\{u\}} \prod_{j=i+1}^{D} \rho_j\right), \quad \mathrm{O}\left(M^2 + \sum_{i=1}^{D} m_i m_i^{\{u\}^2}\right), \tag{3.16}$$

operations with $\mathrm{O}\left(M\right)$ and $\mathrm{O}\left(M^2\right)$ space respectively. The cost of computing and storing the economical Gram matrix Σ in the form of Eq. 3.15 dominates when $M > \max(m_i)$ and is not suitable when some dimensions are not sparse.

Fortunately for **Case 3**, we can express the economical Gram matrix Σ as products of KTPs with diagonal scaling by a reformulation of the inverse of matrix summations. One method is to compute the eigendecompositions of KTP matrices $K_{uu} = \otimes_{i=1}^{D} U_i Z_i U_i^T$ where $U = \otimes_{i=1}^{D} U_i$ and $Z = \otimes_{i=1}^{D} Z_i$ followed by a second set of eigendecompositions of the KTP matrix $Z^{-1/2} U^T K_{uf} K_{fu} U Z^{-1/2} = \otimes_{i=1}^{D} \bar{U}_i \bar{Z}_i \bar{U}_i^T$. This relies on the fact that both matrices K_{uu} and $\sigma^{-2} K_{uf} K_{fu}$ can be fully expressed as KTP eigendecompositions given their original KTP factorizations. Economical matrix Σ can now be expressed as products of KTPs with diagonal scaling given by

$$\Sigma = \sigma^2 \Omega (\bar{Z} + \sigma^2 I)^{-1} \Omega^T, \quad \Omega = U Z^{-1/2} \bar{U}, \quad \bar{U} = \otimes_{i=1}^{D} \bar{U}_i, \quad \bar{Z} = \otimes_{i=1}^{D} \bar{Z}_i, \tag{3.17}$$

which requires

$$\mathrm{O}\left(\sum_{i=1}^{D} m_i^{\{u\}^2} (m_i^{\{u\}} + m_i)\right), \quad \mathrm{O}\left(\sum_{i=1}^{D} m_i^{\{u\}} (m_i^{\{u\}} + m_i)\right), \tag{3.18}$$

operations and space respectively. While matrix Ω is not orthogonal and matrix $(\bar{Z} +$

$\sigma^2 I)^{-1}$ is not the scaled eigenvalues of Σ, the determinant $|\Sigma|$ remains easy to compute as the product of orthogonal matrix determinants cancel to give the expression

$$\log|\Sigma| = \log\sigma^2 + \log|Z| - \log|(Z + \sigma^2 I)|. \tag{3.19}$$

See Appendix 3.9.3 for the derivation of the data LMH and gradients for hyperparameter training. Thus, the overall costs in terms of respective operations and space are given by

$$\mathrm{O}\left(\sum_{i=1}^{D} m_i^{\{u\}^2}(m_i^{\{u\}} + m_i) + \prod_{i=1}^{D} m_i^{\{u\}} + N\sum_{i=1}^{D} m_i^{\{u\}} \prod_{j=i+1}^{D} \rho_j\right), \quad \mathrm{O}\left(\sum_{i=1}^{D} m_i^{\{u\}}(m_i^{\{u\}} + m_i)\right). \tag{3.20}$$

Note that the decomposition by Eq. 3.17 does not apply to other sparse GPR methods such as the fully independent training conditional (FITC) and partially independent training conditional (PITC) where $\wedge_{\text{FITC}} = \mathbf{diag}\,(K_{ff} - Q_{ff} + \sigma^2 I)$, $\wedge_{\text{PITC}} = \mathbf{blockdiag}\,(K_{ff} - Q_{ff} + \sigma^2 I)$, $c_{\text{FITC}} = c_{\text{PITC}} = K_{**}$. This is because matrix $K_{uf}\wedge^{-1} K_{fu}$ within the economical Gram matrix Σ may not have a D-KTP decomposition for a non-constant diagonal in matrix $K_{ff} - Q_{ff}$ subject to arbitrary noise term σ. Thus, we do not extend grid GPR conditions to the FITC and PITC cases.

3.4 Missing Data for Grid GPR

Efficient handling of missing data arise in practical applications where samples may be corrupted by noise and discarded. For tensor datasets, the subsequent loss of a few samples from the training dataset would violate the multidimensional grid conditions; revert-

ing back to standard GPR is not feasible due to the size of the dataset. Fortunately, we show that grid GPR can still perform fast and exact inference and hyperparameter training (compared to standard grid GPR) for small missing data sets $r \in \mathbb{N}^R$ of size R (r_i^{th} row-columns are missing from covariance matrix C). For efficient handling of missing data for sparse-grid GPR, see Appendix 3.9.4.

Formally, let missing observations $y_r \in \mathbb{R}^R$ in $y \in \mathbb{R}^N$ have the corresponding missing input set $X^{\{r\}}$; in the singleton case ($R = 1$), removing a single row-column c_r from C clearly invalidates its KTP decomposition. However, we can express this single row-column deletion within matrix $\hat{C} = C + \sigma^2 I$ as

$$\hat{C} = \begin{bmatrix} C_{11} & c_{1r} & C_{13} \\ c_{r1}^T & c_{rr} & c_{r3}^T \\ C_{31} & c_{3r} & C_{33} \end{bmatrix} + \sigma^2 I, \quad \bar{C} = \begin{bmatrix} C_{11} + \sigma^2 I & 0 & C_{13} \\ 0^T & 1 & 0^T \\ C_{31} & 0 & C_{33} + \sigma^2 I \end{bmatrix}, \tag{3.21}$$

by zeroing out r^{th} row-column and replacing the diagonal entry with 1 in the resulting matrix $\bar{C}$ [92]. The implications are as follows: the determinant of matrix $\bar{C}$ and the entries excluding the r^{th} row-column of the inverse matrix $\bar{C}^{-1}$ would be equivalent to that of a row-column deleted matrix C and its inverse; this is easy to see as row-columns of matrices $\bar{C}$ and $\hat{C}$ may be permuted into dense and identity blocks before carrying out the inversion. Generalizing the validity of the singleton case (Eq. 3.21) for multiple row-column deletions, a transformation from matrix C to $\bar{C}$ can be expressed as rank-1

updates/downdates given by

$$\begin{aligned}
\bar{C} &= \hat{C} + aa^T - bb^T, \\
a &= \sqrt{\frac{||\hat{c}_r||}{2}}\left(\frac{\hat{c}_r}{||\hat{c}_r||} + e_r\right), \quad b = \sqrt{\frac{||\hat{c}_r||}{2}}\left(\frac{\hat{c}_r}{||\hat{c}_r||} - e_r\right), \\
\hat{c}_r &= \left[\begin{array}{c} -c_{1r}, \frac{1-c_{rr}-\sigma^2}{2}, -c_{3r} \end{array}\right]^T,
\end{aligned} \tag{3.22}$$

where vector e_r is the r^{th} column of the identity matrix. The procedure is generalized for multiple R missing inputs in Algorithm 3.

For multiple R missing inputs, the column-vectors b, a can either concatenate into R pairs of successive rank-1 downdates and updates or alternatively into two rank-R downdates and updates $\bar{C} = \hat{C} - BB^T + AA^T$ for $A, B \in \mathbb{R}^{N \times R}$; columns of matrices A and B follow Eq. 3.22 for each vector $\hat{c}_{r_i}$ and zeroing out entries $A_{r_i, i+1:R}$ and $B_{r_i, i+1:R}$. For descriptive purposes, we derive the matrix inversion of $\bar{C}$ from the latter formulation in two steps. The inverse of a rank-R downdate to a matrix $\hat{C}$ [80] and the log-term can be efficiently computed by the modified *Woodbury* formulation for diagonal matrix D given

Algorithm 3 Compute column-vectors to handle row-column r_i deletion (**ComputeABC**)

Require: General column indices $r \in \mathbb{N}^i$, Noise term σ

1: global KTP matrices $[C_1 \in \mathbb{R}^{m_1 \times m_1}, \ldots C_D \in \mathbb{R}^{m_D \times m_D}]$
2: $\hat{c} \leftarrow -\textbf{VKTP}\,(X, x_{r_i} \in X)$ \\ Compute via Algorithm 2
3: $\hat{c}_{r_i} \leftarrow (1 + \hat{c}_{r_i} - \sigma^2)/2$
4: $\hat{c}_{r_{1:i-1}} \leftarrow 0$ \\ Zero-out previous missing data entries
5: $a = (\hat{c}/||\hat{c}|| + e_{r_i})\sqrt{||\hat{c}||/2}$ \\ e_{r_i} is the ${r_i}^{th}$ column of the identity matrix
6: $b = (\hat{c}/||\hat{c}|| - e_{r_i})\sqrt{||\hat{c}||/2}$
7: **return** $a, b, \hat{c} \in \mathbb{R}^N$ \\ Vectors for Eq. 3.22

by

$$
\begin{aligned}
(\hat{C} - BB^T)^{-1} &= \hat{C}^{-1} + B^{(R)} D B^{(R)^T}, \quad B^{(k)} = \left[B_1^{(1)}, \ldots, B_k^{(k)} \right] \in \mathbb{R}^{N \times k}, \\
D_{ii}^{(k)} &= (1 - < B_i, B_i^{(i)} >)^{-1}, \quad \log|\hat{C} - BB^T| = \log|\hat{C}| - \log|D|,
\end{aligned}
\tag{3.23}
$$

where superscript refers to the iteration and subscript the general column index. The column update rule for the desired matrix $B^{(R)}$ is given by

$$
B_{k+1}^{(k+1)} = \left(\hat{C}^{-1} + B^{(k)} D^{(k)} B^{(k)^T} \right) B_{k+1} \in \mathbb{R}^N. \tag{3.24}
$$

The rank-R update to matrix $\bar{C}^{-1} = [(\hat{C} - BB^T) + AA^T]^{-1}$ following the initial rank-R downdate by Eqs. 3.23 and 3.24 has the analogous formulation given by

$$
\begin{aligned}
\bar{C}^{-1} &= \hat{C}^{-1} + B^{(R)} D B^{(R)^T} - A^{(R)} E A^{(R)^T}, \quad A^{(k)} = \left[A_1^{(1)}, \ldots, A_k^{(k)} \right] \in \mathbb{R}^{N \times k}, \\
E_{ii}^{(k)} &= (1 + < A_i, A_i^{(i)} >)^{-1}, \quad \log|\bar{C}| = \log|\hat{C} - BB^T| - \log|E|,
\end{aligned}
\tag{3.25}
$$

with the column update rule for matrix A as

$$
A_{k+1}^{(k+1)} = \left(\hat{C}^{-1} + B^{(R)} D B^{(R)^T} - A^{(k)} E^{(k)} A^{(k)^T} \right) A_{k+1} \in \mathbb{R}^N. \tag{3.26}
$$

The column updates to matrices $B^{(R)}$ and $A^{(R)}$ consist of KTVPs by Eq. 3.12 and a series of $N \times R$ sized matrix-vector products making. The asymptotic costs are given by $\mathrm{O}\left(R^2 N + RN \sum_{i=1}^{D} m_i \right)$ operations and $\mathrm{O}\left(RN \right)$ space. Similarly, the inverse matrix-

vector product $\hat{t} = \bar{C}^{-1}\hat{y}$ is given by

$$\hat{t} = \left(\hat{C}^{-1} + B^{(R)}DB^{(R)^T} - A^{(R)}EA^{(R)^T}\right)\hat{y}, \quad \hat{t}_{i\in X^{\{r\}}} = \hat{y}_{i\in X^{\{r\}}} = 0, \tag{3.27}$$

where zeroing-out entries belonging to missing set r gives valid expression to other terms as $\hat{t}^T P\hat{t}$. The trace term $\mathbf{tr}\left(\bar{C}^{-1}P\right)$ has the analogous expansion given by

$$\mathbf{tr}\left(\bar{C}^{-1}P\right) = \mathbf{tr}\left(\hat{C}^{-1}P\right) + \mathbf{tr}\left(DB^{(R)^T}PB^{(R)}\right) - \mathbf{tr}\left(EA^{(R)^T}PA^{(R)}\right) - \sum_{i\in R} P_{ii}, \tag{3.28}$$

which follows Eq. 3.13, the trace of two $R \times R$ matrices in terms of KTP matrix P, and the subtraction of the missing data diagonal entries in matrix P; the asymptotic costs remain unchanged from computing matrices $A^{(R)}$ and $B^{(R)}$. The predictive variance can be computed from the expansion given by

$$\mathbf{cov}\left(f_*\right) = K_{**} - \left(K_{*f} - K_{*r}\right)\left(\hat{C}^{-1} + B^{(R)}DB^{(R)^T} - A^{(R)}EA^{(R)^T}\right)\left(K_{f*} - K_{r*}\right), \tag{3.29}$$

where matrix $K_{*r} \in \mathbb{R}^{*\times N}$ are the missing columns of K_{*f} and elsewhere zero.

3.5 Extra Data for Grid GPR

Efficient handling of extra data arise in applications where multiple gridded inputs of interest can be selected and evaluated (e.g. patches in image processing, regions of interest in geographic information systems). For subsets within a tensor dataset, the presence of

extra non-gridded samples would violate the multidimensional grid conditions. Fortunately, we show that grid GPR can still perform fast and exact inference and hyperparameter training (compared to standard grid GPR). We first consider the case of the union between unstructured (non-gridded) data with a single multidimensional grid data and then the case of the union between two multidimensional grids.

Unstructured extra data: For input sets $X = \{X^{\{s\}}, X^{\{c\}}\}$ where set $X^{\{s\}}$ of size S contain points outside the grid $X^{\{c\}}$, let the block-covariance matrix be given by $\hat{T} = T + \sigma^2 I$ where $T = K(X, X)$. Its inverse can be formulated under the block-matrix inversion lemma given by

$$\begin{aligned} \hat{T} &= \begin{bmatrix} \hat{H} & G^T \\ G & \hat{C} \end{bmatrix}, \quad \begin{matrix} H = K(X^{\{s\}}, X^{\{s\}}), & C = K(X^{\{c\}}, X^{\{c\}}), \\ \hat{H} = H + \sigma^2 I, & \hat{C} = C + \sigma^2 I, \end{matrix} \\ \hat{T}^{-1} &= \begin{bmatrix} \bar{H} & -\bar{H} G^T \hat{C}^{-1} \\ -\hat{C}^{-1} G \bar{H} & \hat{C}^{-1} G \bar{H} G^T \hat{C}^{-1} + \hat{C}^{-1} \end{bmatrix}, \quad G = K(X^{\{c\}}, X^{\{s\}}), \end{aligned} \tag{3.30}$$

where matrix $\bar{H} = (\hat{H} - G^T \hat{C}^{-1} G)^{-1}$ and the columns of matrix G are VKTPs defined over sets X and $X^{\{s\}}$.

As the extra data size S grows, the cost of the matrix inversion $\bar{H}$ dominates with $\mathrm{O}(S^3)$ operations and can be interpreted as performing standard GPR over the arbitrary input set $X^{\{s\}}$. Computing the inverse matrix-vector product $t = \hat{T}^{-1} y$ via Eq. 3.30 requires a KTVP followed by series of $N \times S$ sized matrix-vector products. Since matrix $\hat{C}$ is invertible, the block-determinant can be expressed as $\log|\hat{T}| = \log|\hat{C}| - \log|\bar{H}|$. The gradient terms $t^T P t$ and $\mathbf{tr}\left(\hat{T}^{-1} P\right)$ are computed from the block partial-derivatives

of matrix $\hat{T}$ and block-matrix products

$$P = \begin{bmatrix} \frac{\partial \hat{H}}{\partial \Theta} & \frac{\partial G^T}{\partial \Theta} \\ \frac{\partial G}{\partial \Theta} & \frac{\partial \hat{C}}{\partial \Theta} \end{bmatrix}, \quad \begin{aligned} \mathbf{tr}\left(\hat{T}^{-1}P\right) &= \mathbf{tr}\left(\bar{H}P_{11}\right) + \mathbf{tr}\left(\hat{C}^{-1}P_{22}\right) \\ &+ \mathbf{tr}\left(G^T\hat{C}^{-1}(P_{22}\hat{C}^{-1}G - 2P_{21})\bar{H}\right) \end{aligned},$$

where the matrix product $\hat{C}^{-1}G$ is expanded using Eq. 3.12 and computed as $2S$ KTVPs.

Gridded extra data: If the extra data set $X^{\{s\}}$ is also a Cartesian outer product with D dimensions, then the covariance matrix $\hat{H}$ and inverse have analogous Kronecker decompositions to that of matrix $\hat{C}$. Difficulties arise in efficiently handling the noise term σ as the eigenvectors of the block-matrix T are not computed and may not be expressible as a single KTP. In the case where the noise term is zero, matrix $\bar{H} = (H + G^T C^{-1} G)^{-1}$ can readily be expressed as products of KTPs with diagonal scaling via Eq. 3.17 by substituting matrices $H \rightarrow K_{uu}$, $G^T C^{-1} G \rightarrow K_{uf} K_{fu}$ and removing the σ term; all block matrices within $\hat{T}^{-1}$ have KTP structures and the total computational costs are the sum of individual costs for two grid GPs specified on inputs $X^{\{c\}}$ and $X^{\{s\}}$.

3.6 Fast Greedy Backward Subset Selection

Classical subset selection methods are commonly used for feature extraction and data reduction prior to classification and regression tasks. One popular method is the GBSS [54] (Algorithm 4) for ranking the input samples as either "salient" or "redundant". GBSS begins with the set of all inputs and progressively removes the least promising ones during each iteration according to an objective function. The objective function is chosen to be grid GPR's remaining data LMH (Eqs. 3.6, 3.25) which can be interpreted as a

measure of similarity between the data and the GP prior assumptions; redundant samples will fit the GP prior assumptions (high LMH) whereas salient ones do not. Conversely, input samples, whose removal would minimize the remaining data LMH, are identified as redundant; inputs that maximize the remaining LMH are identified as salient. Moreover, computing the LMH on the remaining dataset is efficient as the eliminated subset can be treated as missing data from a grid GP (see section 3.4); each input in the remaining dataset is thus tested for its "inclusion" into the missing subset by GBSS during each iteration.

Algorithm 4 Greedy Backward Subset Selection Wrapper (**GBSS**)

Require: Kronecker factors $[C_1 \in \mathbb{R}^{m_1 \times m_1}, \dots C_D \in \mathbb{R}^{m_D \times m_D}]$, Number of missing inputs R, Mode $k = \{-1, 1\}$ (redundant or salient), Observations $y \in \mathbb{R}^N$, Noise term σ

$N \leftarrow \prod_{i=1}^{D} m_i$
$r \leftarrow \emptyset$ \\ Initial subset of inputs
for $i = 1$ to R **do**
  $l \leftarrow -k\infty$ ones$(N, 1)$ \\ Initial LMH
  for $\hat{r} \notin r$ **do**
    $l_{\hat{r}} \leftarrow$ **TestCol**$(i,$ **ColToK**$(\hat{r}), [r, \hat{r}], y, \sigma)$ \\ Data LMH
  end for
  $r \leftarrow [r, \arg\max(kl)]$ \\ Select minimizing or maximizing input
  UpdateCol$(i,$ **ColToK**$(r_i), r, \sigma)$
end for
return $r \in \mathbb{N}^R$

For notation, let input r be the missing dataset of size R and $r_i \in r$ be the i^{th} input removed. To express the R number of row-column deletions to the inverse covariance matrix $\bar{C}^{-1}$ (Eq. 3.22), it is more efficient to use the formulation of R pairs of successive rank-1 updates/downdates than the formulation of two rank-R updates/downdates (Eqs. 3.23 and 3.25). The former modifies the order for computing the leading i^{th} column of matrices $B^{(R)}, D^{(R)}, A^{(R)}, E^{(R)}$ once an input is included (see function **UpdateCol** in

Algorithm 5); the latter modifies all of matrix $A^{(R)}$ for any point changes made to matrix $B^{(R)}$ which is inefficient. This allows the i^{th} row-column rank-1 update/downdate to $\hat{C}^{-1}$, given by

$$\xi^{(i)} = \hat{C}^{-1} - A^{(i)}_{:,1:i-1} E^{(i)}_{1:i-1} A^{(i)^T}_{:,1:i-1} + B^{(i)}_{:,1:i-1} D^{(i)}_{1:i-1} B^{(i)^T}_{:,1:i-1}, \tag{3.31}$$

to be efficiently computed in $\mathrm{O}\left(N(\sum_{j=1}^{D} m_j) + Ni + N^2\right)$ operations and $\mathrm{O}\left(Ni + N^2\right)$ space.

Another computational improvement follows from modifying GBSS to reuse results from inclusion tests from prior iterations; the test for the inclusion of input $\hat{r}$ between two successive iterations can be made efficient. Consider the two sequences of missing inputs evaluated at rounds i and $i+1$ where $\hat{r}$ is evaluated last:

$$r^{(i)} = [r_{1:i-1}, \hat{r}], \quad r^{(i+1)} = [r_{1:i}, \hat{r}], \tag{3.32}$$

where differ by the inclusion of input r_i; both sequences are preconditions for all calls

Algorithm 5 Update column i of matrices $B^{(i)}, D^{(i)}, A^{(i)}, E^{(i)}$ (**UpdateCol**)

Require: i^{th} column update for D-KTP column index $\bar{q} = \textbf{ColToK}(r_i) \in \mathbb{N}^D$, Missing data set $r \in \mathbb{N}^i$, Observations $y \in \mathbb{R}^N$, Noise term σ

1: global $B^{(i)}, A^{(i)} \in \mathbb{R}^{N \times i}, D^{(i)}, E^{(i)} \in \mathbb{R}^i, \xi \in \mathbb{R}^{N \times N}$ \\ Missing-set internals
2: $[a, b] \leftarrow \textbf{ComputeABC}(r, \sigma)$ \\ Algorithm 3
3: $B^{(i)}_{:,i} \leftarrow \left(\hat{C}^{-1} - A^{(i)}_{:,1:i-1} E^{(i)}_{1:i-1} A^{(i)^T}_{:,1:i-1} + B^{(i)}_{:,1:i-1} D^{(i)}_{1:i-1} B^{(i)^T}_{:,1:i-1}\right) b$
4: $D^{(i)}_i \leftarrow 1/(1 - < B^{(i)}_{:,i}, b >)$ \\ Update diagonal by Eq. 3.23
5: $A^{(i)}_{:,i} \leftarrow \left(\hat{C}^{-1} - A^{(i)}_{:,1:i-1} E^{(i)}_{1:i-1} A^{(i)^T}_{:,1:i-1} + B^{(i)}_{:,1:i} D^{(i)}_{1:i} B^{(i)^T}_{:,1:i}\right) a$
6: $E^{(i)}_i \leftarrow 1/(1 + < A^{(i)}_{:,i}, a >)$ \\ Update diagonal by Eq. 3.25
7: $\xi^{(i)} \leftarrow \xi^{(i-1)} - A^{(i)}_{:,i-1} E^{(i)}_{i-1} A^{(i)^T}_{:,i-1} + B^{(i)}_{:,i-1} D^{(i)}_{i-1} B^{(i)^T}_{:,i-1}$ \\ Initial $\xi^{(0)} = \hat{C}^{-1}$

made to **TestCol** (Algorithm 6) until test input $\hat{r}$ is added into the missing set. The rank-1 column update-downdate vectors a, b, c (Eq. 3.22, function **ComputeABC** for Algorithm 3) can be specified as a set of efficient recurrence-relations: The difference-vector $c^{(i+1)} - c^{(i)} = -c^{(i)}_{r_i} e_{r_i}$ is simply a sparse vector with a single non-zero entry at index r_i. This allows column-vectors c to be expressed as two equivalent recurrence relations such that a, b are efficiently computed:

$$
\begin{aligned}
c^{(i)} &= a^{(i)}\sqrt{2|c^{(i)}|} - |c^{(i)}|e_{\hat{r}}, \quad a^{(i+1)} = \frac{-c^{(i)}_{r_i} e_{r_i} + (|c^{(i+1)}| - |c^{(i)}|)e_{\hat{r}}}{\sqrt{2|c^{(i+1)}|}} + a^{(i)}\sqrt{\frac{|c^{(i)}|}{|c^{(i+1)}|}}, \\
c^{(i)} &= b^{(i)}\sqrt{2|c^{(i)}|} + |c^{(i)}|e_{\hat{r}}, \quad b^{(i+1)} = \frac{-c^{(i)}_{r_i} e_{r_i} - (|c^{(i+1)}| - |c^{(i)}|)e_{\hat{r}}}{\sqrt{2|c^{(i+1)}|}} + b^{(i)}\sqrt{\frac{|c^{(i)}|}{|c^{(i+1)}|}},
\end{aligned}
\tag{3.33}
$$

where updates to $a^{(i+1)}, b^{(i+1)}$ are vector-summations of the preceding column vector (scaled) and a sparse column (two non-zero entries at indices r_i and $\hat{r}$). This formulation (Eq. 3.33) allows the subsequent matrix-columns $B^{(i+1)}_{i+1}$ and $A^{(i+1)}_{i+1}$ to be updated in-place within function **TestCol** for each test input $\hat{r}$ and stored in matrix-columns $\hat{B}_{:,\hat{r}}$ and $\hat{A}_{:,\hat{r}}$ respectively. The column update operations are efficient as they only use vector scaling, two vector summations (made possible by an invariant matrix $\xi^{(i)}$ between calls to **TestCol**), and six rank-1 matrix-vector products. Thus, the costs of one GBSS iteration are reduced to O (N) operations and O (N^2) space.

Algorithm 6 Test inclusion of input $r_i = \hat{r}$ into missing data set (**TestCol**)

Require: i^{th} inclusion test for D-KTP column index $\bar{q} = \textbf{ColToK}(r_i) \in \mathbb{N}^D$, Missing data set $r \in \mathbb{N}^i$, Observations $y \in \mathbb{R}^N$, Noise term σ

global $B^{(i)}, A^{(i)} \in \mathbb{R}^{N \times i}, D^{(i)}, E^{(i)} \in \mathbb{R}^i, \xi^{(i-1)} \in \mathbb{R}^{N \times N}$ \\ Missing-set internals

global $\hat{B}, \hat{A}, \hat{T} \in \mathbb{R}^{N \times N}$ \\ Test-input internals

\\ Compute current and previous columns using Algorithm 3

$[a^{(i)}, b^{(i)}, c^{(i)}] \leftarrow \textbf{ComputeABC}(r, \sigma)$

$[c^{(i-1)}] \leftarrow \textbf{ComputeABC}([r_{1:i-2}, r_i], \sigma)$

$\hat{b} \leftarrow -\frac{\left[c_{r_i}^{(i-1)};\quad ||c^{(i)}||-||c^{(i-1}||)\right]}{\sqrt{2||c^{(i)}||}}, \quad \hat{a} \leftarrow -\frac{\left[c_{r_i}^{(i-1)};\quad ||c^{(i-1)}||-||c^{(i)}||\right]}{\sqrt{2||c^{(i)}||}} \in \mathbb{R}^2$

\\ Update by Eq. 3.33 and latest rank-1 update-downdate pair

$\beta \leftarrow \sqrt{\frac{||c^{(i-1)}||}{||c^{(i)}||}}\hat{B}_{:,r_i} + \xi^{(i-1)}_{:,r_{i-1:i}}\hat{b} - (A^{(i)}_{:,i-1}E^{(i)}_{i-1}A^{(i)^T}_{:,i-1} - B^{(i)}_{:,i-1}D^{(i)}_{i-1}B^{(i)^T}_{:,i-1})b^{(i)}$

$\delta \leftarrow 1/(1- < \beta, b^{(i)} >)$ \\ Diagonal entry in Eq. 3.23

$\alpha \leftarrow \sqrt{\frac{||c^{(i-1)}||}{||c^{(i)}||}}\hat{A}_{:,r_i} + \xi^{(i-1)}_{:,r_{i-1:i}}\hat{a} - (A^{(i)}_{:,i-1}E^{(i)}_{i-1}A^{(i)^T}_{:,i-1} - B^{(i)}_{:,i-1}D^{(i)}_{i-1}B^{(i)^T}_{:,i-1})a^{(i)}$

$\hat{B}_{:,r_i} \leftarrow \beta, \quad \hat{A}_{:,r_i} \leftarrow \alpha$ \\ Write columns to internal matrices

$\alpha \leftarrow \alpha + \beta\delta\beta^T a^{(i)}, \quad \gamma \leftarrow 1/(1+ < \alpha, a^{(i)} >)$ \\ Diagonal entry in Eq. 3.23

$y_r \leftarrow 0, \quad t \leftarrow \hat{T}_{:,r_i} - \xi^{(i-1)}_{:,r_{i-1}}y_{r_{i-1}} - (A^{(i)}_{:,i-1}E^{(i)}_{i-1}A^{(i)^T}_{:,i-1} - B^{(i)}_{:,i-1}D^{(i)}_{i-1}B^{(i)^T}_{:,i-1})y$

$\hat{T}_{:,r_i} \leftarrow t, \quad t \leftarrow t + (\beta\delta\beta^T - \alpha\gamma\alpha^T)y$

$l \leftarrow 1/2\left(\log|C| - \sum_{j=1}^{i-1}(\log D_j^{(i)} + \log E_j^{(i)}) - \delta - \gamma + < y, t > + (N-i)\log(2\pi)\right)$

return $-l \in \mathbb{R}$ \\ Remaining data LMH

3.7 Experiments and Applications

GP models (standard, sparse, grid, sparse-grid) are specified and trained on both synthetically generated high-dimensional tensor data and real 2D-HRTF datasets. GP covariance function hyperparameter optimization use the natural gradient (Eq. 3.6) with resilient back-propagation (RPROP) [93]; RPROP locally rescales each hyperparameter via an online step-size adaptation based the sign of the gradient (evaluated once per iteration). The heuristic gives a fast convergence rate and prevents oscillatory behavior compared to standard gradient descent methods. Compared to nonlinear conjugate gradient, also RPROP provides tractable run-time cost analyses due to the absence of line-search. All experiments are conducted on an Intel i7-2630QM laptop running Matlab 2010 on 64-bit

Windows.

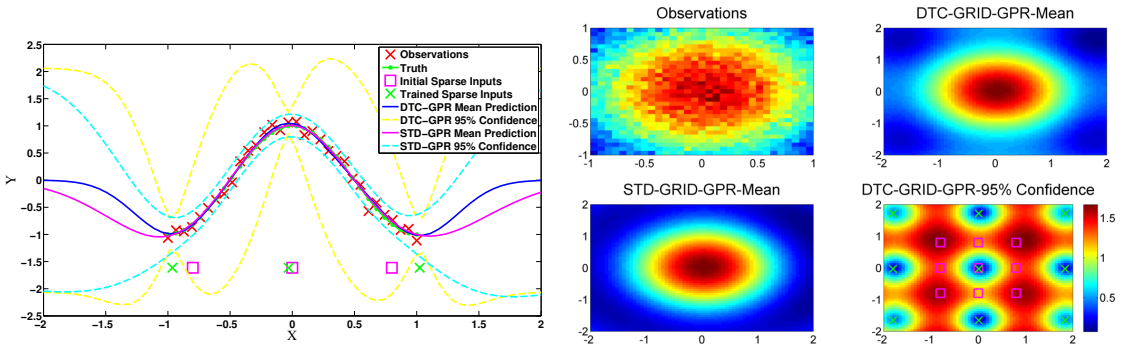
3.7.1 Performance Tests on Synthetic Data

The performance (training and inference runtimes) gains by grid and sparse-grid GP methods over standard GP and sparse GP is easily demonstrated in the case of high multi-dimensional gridded data. Consider the following toy-example: let a D dimensional cube centered about the origin be uniformly partitioned into a multi-dimensional grid of inputs $X = X_1 \times \ldots \times X_D$ with linear spacing of size $m = 32$ along each dimension. Let the $N = m^D$ number of inputs index the respective outputs y which are initialized to the input vector's Euclidean norm and then mapped to the angular frequency of the cosine function given by

$$X_j = \left\{ -1 : \frac{2}{m-1} : 1 \right\}, \quad y_i = \cos\left(\sqrt{\frac{1}{D} \sum_{j=1}^{D} x_{ij}^2} \right), \tag{3.34}$$

before corrupted by additive Gaussian white noise $\hat{y}_i = y_i + \mathcal{N}(0, \sigma^2)$ where $\sigma = .3$. The underlying function y is convex in the domain of X such that for sparse-grid GPR, we can specify a low number of inducing points ($m^{\{u\}} = 3 \rightarrow m$) without overfitting; the trained inducing points should ideally be equidistant from the origin. The well-known squared exponential covariance function $K_i(x_j, x_k) = \exp\left(-\frac{(x_j - x_k)^2}{2\theta_i^2} \right)$ is specified along the i^{th} dimension such that the overall covariance is the TPK covariance $\mathbf{cov}\,(x_j, x_k) = \alpha^2 \prod_{i=1}^{D} K_i(x_j, x_k)$. The global-scale hyperparameter α and D number of length-scale hyperparameters θ_i are trained for 50 iterations. Figure 3.1 illustrates posterior means and variances at a gridded test set of inputs on 2D synthetic data by standard (STD) grid

and sparse grid GPs; the posterior variances grow larger for locations further from the inducing locations.

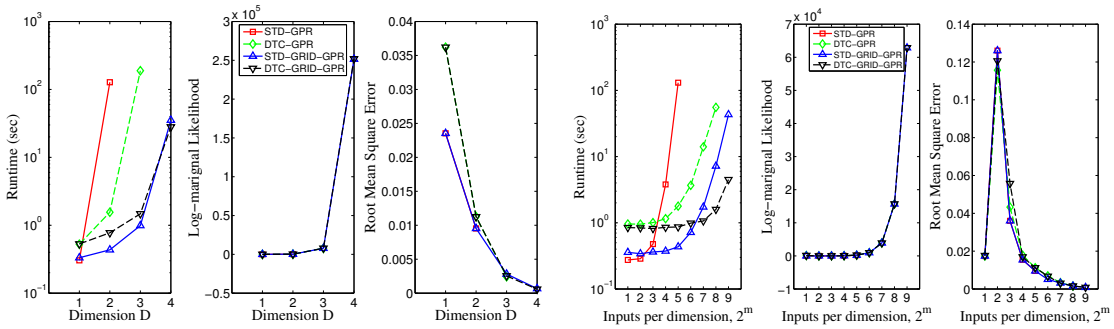


(a) 1D Posterior Distribution (b) 2D Posterior Distribution

Figure 3.1: Posterior distributions for standard grid and sparse-DTC grid GP methods are shown for synthetic data (Eq. 3.34). Initial sparse inputs (□) once trained (X) move away from the origin in the 2D case.

Varying D and input size m per dimension: Synthetic high-dimensional tensor datasets (Eq. 3.34) are generated for two cases of an increasing input dimension D with a fixed number of samples m per dimension, and an increasing m with fixed D. Standard, sparse, standard grid, and sparse grid GPs are specified and trained on this dataset. Their prediction accuracy is evaluated by the root mean squared error (RMSE) metric ($\sqrt{\sum_{i=1}^{N}(\bar{f}_i - y_i)^2/N}$ at training input x_i) between the posterior means predictions and reference observations. Figure 3.2 compares the models' training runtime, data LMH, and the predictions' RMSE. The results are expected as sparse-grid GPR scales better than standard grid GP training for fixed dimension D and increasing input sizes N as the number of inducing points to train remain constant. Standard and sparse GPs did not terminate for $D > 3$ due to memory and runtime restrictions and similarly for m greater than 2^5 and 2^7 respectively for fixed dimension $D = 2$.

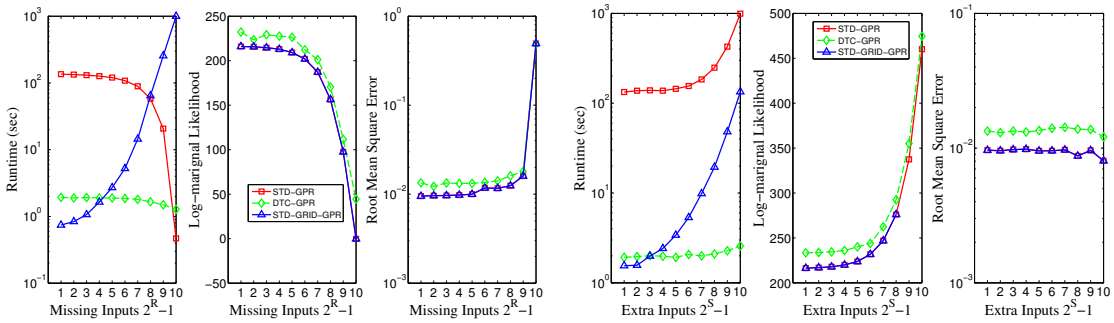
Varying number of missing data R and number of extra data S: Synthetic



(a) Varying dimension D (b) Varying inputs per dimension

Figure 3.2: Training runtimes, LMH, and RMSE are shown for cases of varying dimension D (fixed $m = 32$, $M = 32^D$) and varying number of inputs per dimension 2^m (fixed dimension $D = 2$, $M = (2^m)^2$) for standard, DTC, standard grid, and sparse-DTC grid GP methods.

datasets (Eq. 3.34) are generated for fixed $D = 2$ and $m = 32$. In one case, a variable number R of samples are removed from the dataset to simulate missing data. In another case, a variable number S of samples are added (in accordance with Eq. 3.34) to the dataset to simulate extra data. The number of missing and extra inputs are incremented from $2^{1\ldots10} - 1$ in powers of 2; the full range of missing inputs (from one to all but one) and extra data (from one to near the size of original inputs) are covered. Standard, sparse, and standard grid GPs are specified and trained on these datasets; grid methods use efficient techniques for handling both cases of missing and extra data. Figure 3.3 compare the models' training runtime, data LMH, and predictions' RMSE. The runtimes for standard and grid GPR in missing data crossover after a quarter of the inputs $R = 2^8 - 1$ are missing. The runtimes for standard and grid GPR in extra data converge as the cubic runtime costs dominate. We omit the implementation for the case of missing and extra data in sparse-grid GPR due to limited applications.



(a) Varying missing data R (b) Varying extra data S

Figure 3.3: Training runtimes, LMH, and RMSE are shown for cases of varying number of missing data R and number of extra data S for fixed dimension $D = 2$ and fixed number of inputs (32^2) across standard, DTC, and standard grid GP methods.

3.7.2 Grid and Sparse-Grid GPs for HRTF Interpolation

Spatial-temporal datasets can often be parameterized by Cartesian outer products between the location and time of measurements; this is a common occurrence for time-series measurements collected by a collection of fixed sensors over a region. Such is the case for HRIR/HRTF datasets (e.g. CIPIC [1]), which consists of acoustic time-series measurements by microphones placed in the human ears that record broad-band test signals (impulses) emitted by loudspeakers positioned along a spherical-coordinate grid. Each measurement, parameterized by a spherical elevation and azimuth pair $X_s = (\theta, \phi)$, contains information on how the sound source's acoustic wave scatters off of a subject's anatomic features (torso, head, and outer ears) before reaching the eardrum. The information can be represented as either a time-series impulse response (HRIR) or by a frequency-domain transfer function (HRTF) which are interchangeable via the Fourier and inverse Fourier transforms [7]. The latter magnitude HRTF representation is useful as the samples are observed to be smooth in both the spatial and frequency (parameterized by wave-number

X_ω) domains. Thus, collections of HRTFs[3], belonging to the same subject, can be parameterized by the Cartesian outer product of spatial-frequency input domains $X = X_s \times X_\omega$ and modeled by stationary covariance functions.

One important application is the reconstruction of life-like auditory scenes via HRTFs; acoustic waves (e.g. direct and reflected sound paths) that would enter the ear from different directions can be simulated by convolving an HRTF with the sound-source. Two issues arise in practice: A finite collection of HRTF measurements will never span the entire spherical coordinate system. The HRTFs may also have different sampling rates than that of the sound-source data. Both problems can be solved by learning an interpolant (i.e. grid GP) between the input spatial-frequencies X and the output magnitude responses $y = |\text{HRTF}(\theta, \phi, \omega)|$.

Under TPK assumptions (separable covariance functions), we specify grid GP's covariance function as the product of the *Ornstein-Uhlenbeck* (OU) [94] spectral density (frequency domain covariance function) and the exponential of the chordal ("great-circle distance") distance[4] (spatial domain covariance function) given by

$$\begin{aligned} K(\theta_i, \theta_j, \phi_i - \phi_j, \omega_i - \omega_j) &= \frac{\alpha^2}{\lambda^2 + (\omega_i - \omega_j)^2} \exp\left(-\frac{C_h(\theta_i, \theta_j, \phi_i - \phi_j)}{\ell^2}\right), \\ C_h(\theta_i, \theta_j, \phi_i - \phi_j) &= 2\sqrt{\sin^2\left(\frac{\theta_j - \theta_i}{2}\right) + \sin\theta_i \sin\theta_j \sin^2\left(\frac{\phi_i - \phi_j}{2}\right)}. \end{aligned} \tag{3.35}$$

The OU process simulates a stochastic differential equation with standard Brownian motion; the λ term refers to the rate of mean reversion (drift to zero) which agrees with the

[3]The CIPIC database consists of 1250 HRTFs measured over a spherical grid for 45 different subject's left and right ears. Each measurement consists of 200 time samples, which after taking the magnitude of its Fourier transform, is reduced to 100 frequency bins.

[4]distance on the unit sphere

observation that HRIRs quickly decay to zero after the initial onset. The chordal distance is selected as it represents a physical distance between two points on the unit sphere. The exponential function is empirically selected via the GP goodness-of-fit criterion (largest data LMH). In theory, valid processes over a sphere must have spatial covariance functions that are expressible in a proper spherical basis [95], e.g.

$$K(\theta_i, \theta_j, \phi_i, \phi_j) = \sum_{n=0}^{\infty} b_n \frac{4\pi}{2n+1} \sum_{m=-n}^{n} Y_n^m(\theta_i, \phi_i) \bar{Y}_n^m(\theta_j, \phi_j), \tag{3.36}$$

for spherical harmonic basis $Y_n^m(\theta, \phi)$ (Eq. 1.5) and coefficients b_n that depend on the choice and parameterization of K (see Appendix 3.9.6). Moreover, the family of isotropic covariances restricted to distances in $\mathbb{R}^3$, such as chordal Euclidean and C_h, are known to be valid on the unit sphere [96, 97]. Thus, the expected realizations of spherical processes (i.e. grid GP's posterior mean function $\bar{f}_*$ in Eq. 3.5) can be expressed along a spherical harmonics basis as they are simply weighted combinations of covariance function evaluations (Representer theorem).

3.7.2.1 Grid and Sparse-Grid GPs Comparisons

Grid and sparse GP models are specified and trained on HRTFs (CIPIC subject 3, right-ear). For sparse-grid GPs, the number of inducing inputs $X^{\{u\}}$ is constrained to be sparse in either the frequency or the spatial domain but not both; the sparse subset of inducing inputs are optimized for 100 iterations while the remaining inducing inputs (on the opposing axis) are fixed. For an illustration, GPs are specified on the collection of azimuth

plane ($\theta = \pi$ radians) HRTFs (50 measurements) and optimized[5]; Figure 3.4 displays the predicted mean response and confidence intervals at evenly spaced test inputs along the spherical-frequency domains.

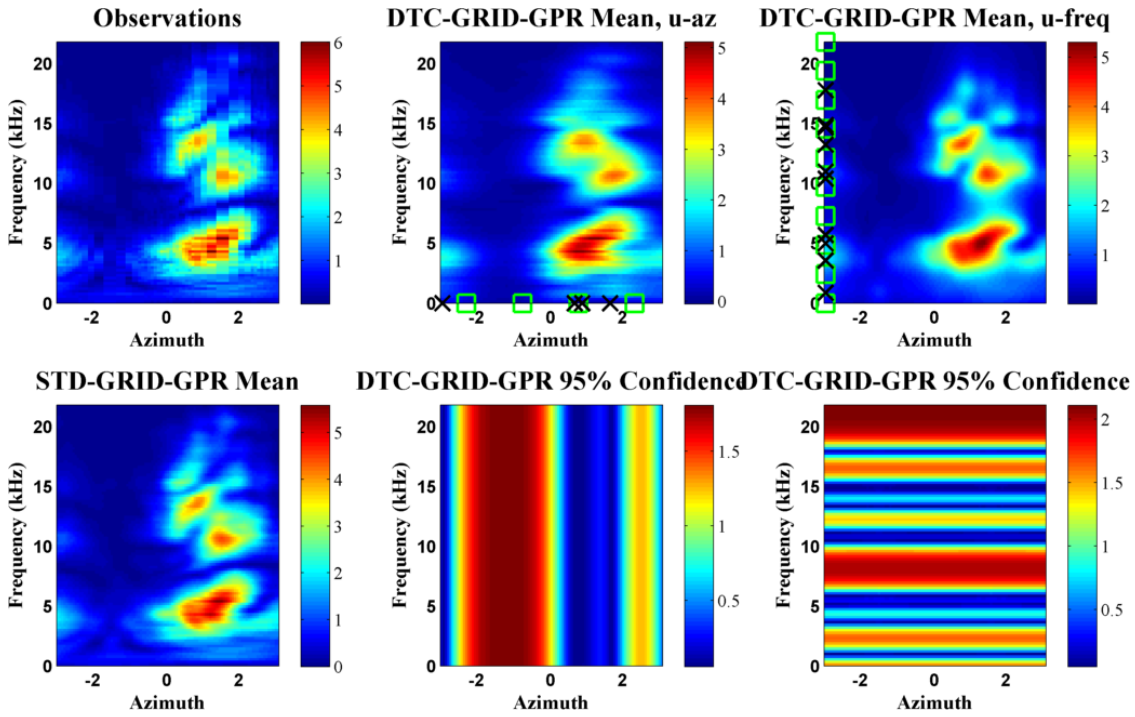


Figure 3.4: Posterior mean magnitude responses and variances for grid and sparse-DTC-grid GPR along the azimuth plane are shown. Initial inducing inputs are marked as □ and trained inputs are marked as **X**.

For large-scale experiments, grid and sparse-grid GPs are trained over the full collection of 1250 HRTFs (covering most of the sphere except an open hole below the head) for each CIPIC subject. We adopt two distance measures to evaluate the predicted magnitude responses: the spectral distortion (SD) is a logarithmic distance measure (dB) between the overall reference and predicted spectra. The signal-to-distortion ratio (SDR)

[5]Sparse grid GP inducing inputs are constrained to be sparse in either frequency (10 of 100 bins) or spatial (4 of 50 directions) domains but not both. Noise term is set to a constant ($\sigma = 0.05$).

provides a per-frequency measure of similarity. Both are given as

$$SD(H(\theta,\phi)) = \sqrt{\frac{1}{|X_\omega|}\sum_{i=1}^{|X_\omega|}\left(20\log_{10}\frac{|H_i(\theta,\phi)|}{|\hat{H}_i(\theta,\phi)|}\right)^2},$$
$$SDR_\omega = 10\log_{10}\frac{\sum_{i=1}^{|X_s|}H_\omega(\theta_i,\phi_i)^2}{\sum_{i=1}^{|X_s|}(H_\omega(\theta_i,\phi_i)-\hat{H}_\omega(\theta_i,\phi_i))^2}, \tag{3.37}$$

where H and $\hat{H}$ are the true and predicted magnitude responses respectively.

Figure 3.5 shows the runtime to SD error trade-off between grid and sparse-grid GP training. For sparse-grid GPs, the inducing inputs in frequency are fixed (set to the full set of inputs X_ω) and the M number of inducing inputs in the spatial dimension are varied; both hyperparameters and inducing inputs are optimized for 50 iterations. Sparse-grid GP's LMH approaches that of grid GPR after $M \geq 150$. The SD error flattens after $M \geq 75$ as further accuracy on the log-scale requires a larger M; most of the error occurs in higher frequency ranges where the magnitude response tends towards 0. The experiment is repeated across the remaining subjects in the CIPIC database. Figure 3.6 shows the overall trade-off between runtime and SD for different frequency intervals. Sparse-grid GP obtains perceptually indistinguishable SD errors (< 3 dB) in the low to mid frequencies ($0 - 18$ kHz) at a fraction of the runtime costs compared to grid GP.

3.7.2.2 Cross-Validation Experiments

In the first experiment, a random half of the 1250 HRTF measurements (subject 3, right-ear) is chosen as the training set. Grid GP models are trained (50 iterations of hyper-parameter optimization) and then predict HRTFs at the hold-out set (remaining inputs);

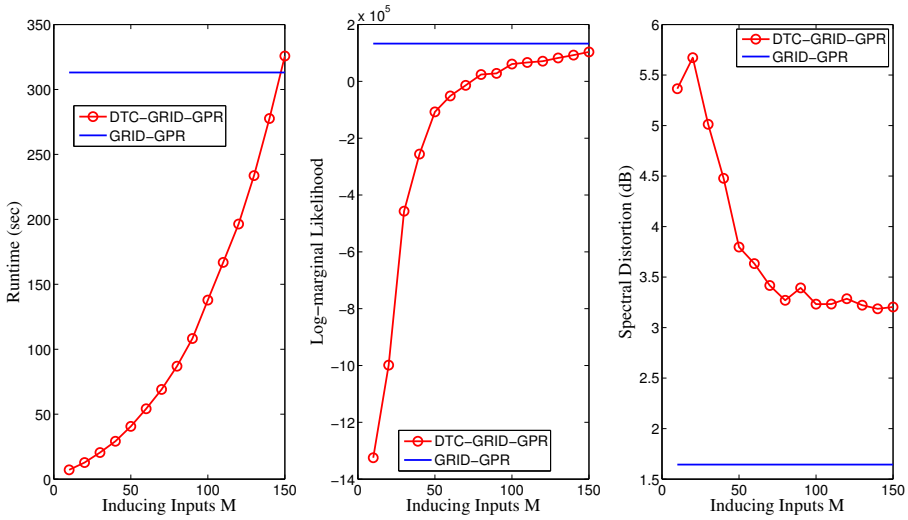


Figure 3.5: Learning curves (runtime, LMH, and SD) for grid and sparse-DTC-grid GP methods for are shown for increasing number of inducing inputs. Lower SD indicates more accurate predictions.

the SDRs (Eq. 3.37) w.r.t. HRTFs in the hold-out set are shown in Figure 3.7(a), which generalize the prediction errors over the entire spherical coordinate system. Other interpolation methods are compared: Inverse distance linearly interpolates HRTFs according to the nearest $k = 4$ measurement directions. Spherical splines [42] fit a Legendre polynomial basis over the sphere (default parameters for smoothing and expansion terms are used). Spherical harmonic fitting [41] finds a least squares solution via a truncated SVD method. The results show that grid GP outperforms (higher SDR) all other methods in the $2 - 20$ kHz frequency range.

In the second experiment, we simulate missing inputs within a large spatial cone (open hole task [98]) by removing all measurements that lie above a horizontal plane (spherical incident angle $\theta < \pi/5$ containing 147 measurement directions). This experiment mimics the problem of inferring HRTFs over large areas where nearby data is unavailable (e.g. the bottom hole in most HRTF measurement grids). Grid GP and other

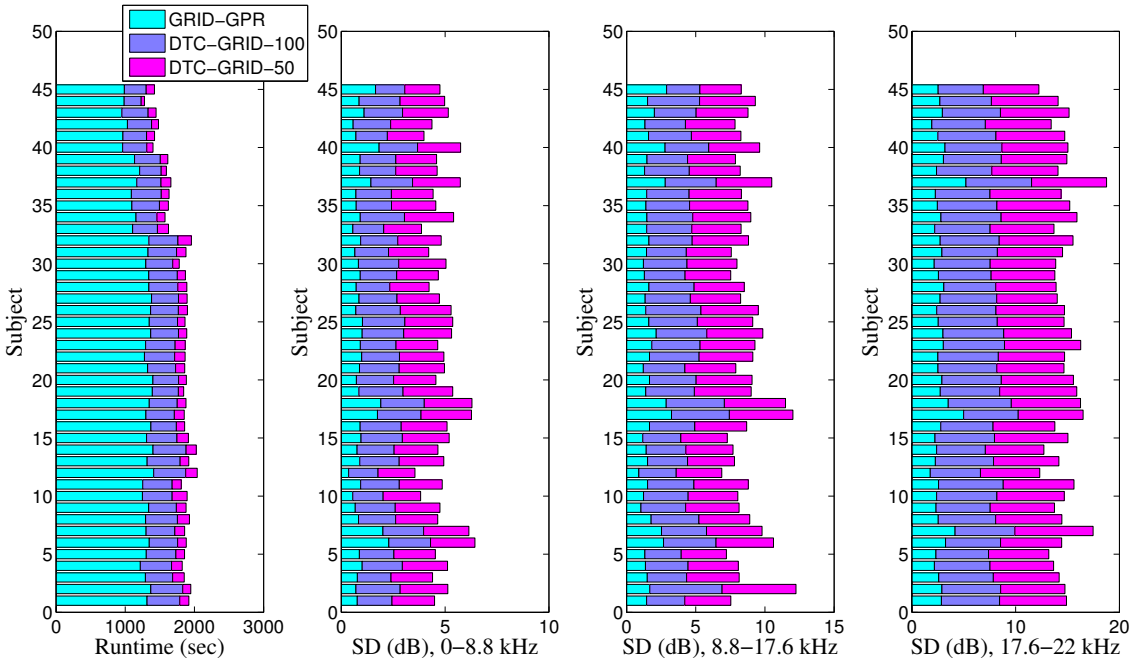
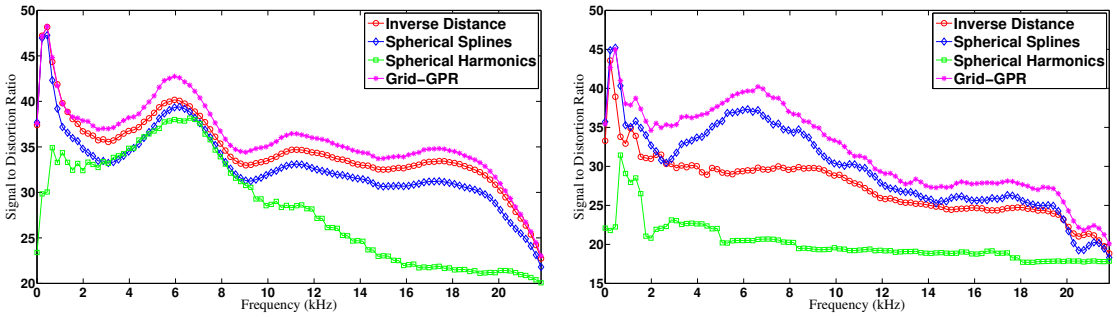


Figure 3.6: Spectral distortion (SD) errors are shown for predictions made by grid and sparse-DTC-grid GPs across 45 CIPIC subject right-ear HRTF datasets. Sparse cases consist of 100 and 50 inducing inputs (optimized in spherical domain, fixed in frequency).

interpolants are trained (50 iterations of hyperparameter optimization) over the remaining measurements and evaluated at the test set. The SDRs are computed and shown in Figure 3.7(b). Grid GPR has the lowest errors along the $2-10$ kHz frequency range and consistently outperforms the other interpolations in the remaining frequencies.

3.7.2.3 Kernel Function Series Expansions

Acoustic measurement apparatus are commonly designed as sensor arrays arranged along one or two fixed axes. For example, $2-3$D microphone arrays have sensors that are aligned/placed onto a rectangular grid; the spatial topology (sensor locations) can naturally be expressed as Cartesian outer products between points along two to three Cartesian coordinate axes. For grid GP, this allows a valid separable TPK to be specified as the



(a) Interpolation Accuracy (b) Extrapolation Accuracy

Figure 3.7: Cumulative SDRs (dB) for the interpolation (random partition) and extrapolation (missing hole) experiments are shown for grid GPR, inverse distance, spherical harmonic, and spline interpolants. Larger SDRs indicate more accurate predictions.

product of covariance functions that are restricted to inputs along each axis. Moreover, some "mixed" covariance functions, such as the squared exponential, can be shown to be separable via series expansion (e.g. power series) laws:

$$\begin{aligned} e^{x} &= \sum_{k=0}^{\infty} \frac{x^k}{k!}, \quad e^{2x_i x_j} = \sum_{k=0}^{\infty} \frac{(2x_i x_j)^k}{k!}, \\ e^{-(x_i - x_j)^2} &= \sum_{k=0}^{\infty} \frac{2^k x_i^k e^{-x_i^2} x_j^k e^{-x_j^2}}{k!}, \end{aligned} \tag{3.38}$$

where the covariance matrix would be approximated as the truncated sum of KTP matrices.

We show that the analogous treatment of spherical covariance functions on spherical measurement-grid inputs is also possible: For azimuth and elevation parameters (θ, ϕ), the squared exponential covariance of the chordal distance C_h (Eq. 3.35) is given by

$$e^{-\frac{C_h^2}{2\ell^2}} = \sum_{k=0}^{\infty} \frac{\left(-2\ell^{-2} \sin\theta_i \sin\theta_j \sin^2\left(\frac{\phi_i - \phi_j}{2}\right)\right)^k e^{-\frac{2\sin^2\left(\frac{\theta_j - \theta_i}{2}\right)}{\ell^2}}}{k!}, \tag{3.39}$$

and thus expressible as a sum of products between θ and ϕ variables. The covariance

matrix K where $K_{ij} = e^{-\frac{C_h(\theta_i,\theta_j,\phi_i,\phi_j)^2}{2\ell^2}}$ is expressed as a truncated (ρ number of terms) sum of KTPs given by

$$\begin{aligned} K &\approx \sum_{k=0}^{\rho} \frac{\left(-2\ell^{-2}\right)^k K_\theta \otimes K_\phi}{k!}, \\ K_\theta &= \left(\sin\theta_i \sin\theta_j\right)^k e^{-\frac{2\sin^2\left(\frac{\theta_j-\theta_i}{2}\right)}{\ell^2}}, \quad K_\phi = \sin^{2k}\left(\frac{\phi_i-\phi_j}{2}\right). \end{aligned} \tag{3.40}$$

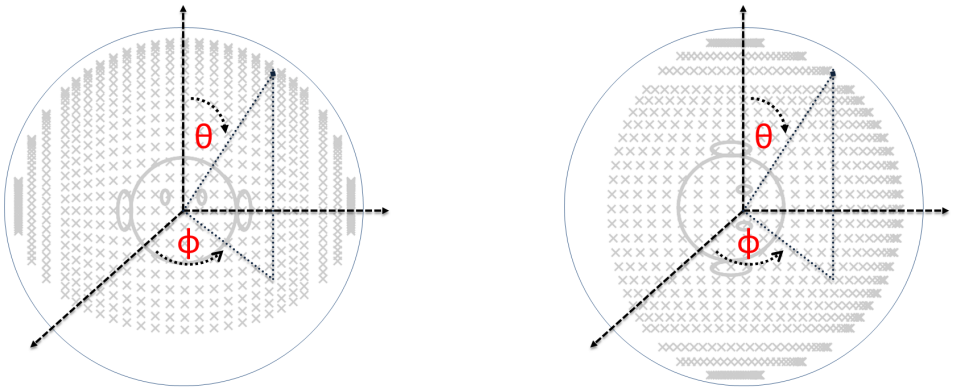
Efficient grid GP inference and training would thus take advantage of subsequent KTVP operations that are enabled by these decompositions.

CIPIC [1] measurement grid analysis: The set of HRIR measurements are recorded over directions corresponding to a rigid hoop of speakers. During recording sessions, the hoop is rotated about the horizontal axis (parallel to the subject's ears) as shown in Figure 3.8. The speaker locations can be made to fall on the grid of spherhical coordinates $\theta \times \phi$ if the original directions are rotated by 90° or if two of the axes along the standard basis are swapped. Note that this would not compromise the stationary kernel function in Eq. 3.39 as the angular distances would remain invariant to rotations of the underlying coordinate system.

The approximation error, due to truncation term ρ, can be bounded by considering the Lagrangian remainder of the kernel function taken w.r.t. C_h^2 and given by

$$R_\rho(C_h^2) = \frac{\left| e^{-\frac{z}{2\ell^2}} \left(-\frac{C_h^2-a}{2\ell^2}\right)^{\rho+1} \right|}{(\rho+1)!} \leq \frac{\left| \left(-\frac{C_h^2}{2\ell^2}\right)^{\rho+1} \right|}{(\rho+1)!}, \tag{3.41}$$

where for center of expansion $a = 0$, the term $z = 0$ gives an upper bound on the remainder for $0 \leq C_h \leq \pi$. Figure 3.9 shows how the error rapidly decays with fewer



(a) CIPIC Measurement Grid (b) Rotated Measurement Grid

Figure 3.8: CIPIC measurement directions are mapped to a spherical coordinate grid under a simple rotation.

truncation terms ρ for large hyperparameter term ℓ and small C_h. For hyperparameter $\ell = 1.0$, a truncation term of $\rho = 13$ has an upper bound (approximation error) of 0.0583 for the maximum $C_h = \pi$.

3.7.2.4 Spectral-Extrema Extraction

Spectral extrema (such as peaks and notches) of magnitude HRTFs have been shown to correlate listening cues along specific directions (median plane) to anatomical features [99, 100]. Extracting the spectral extrema can be done by fitting smooth basis functions to the magnitude spectra (e.g. cosine basis) and finding the local minima and maxima. For the grid GP, the spectral extrema of the predicted HRTFs correspond to the zero-crossing of the posterior mean (Eq. 3.5) function's gradient, which are weighted combinations of smooth covariance functions. The covariance function's first and second-order partial

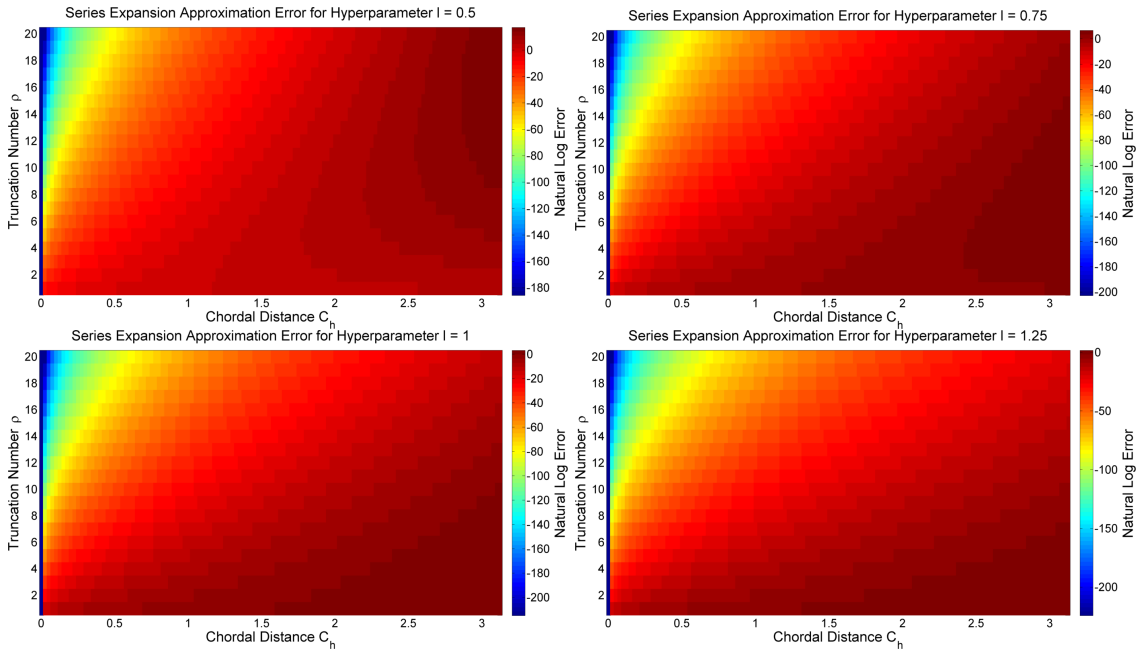


Figure 3.9: Approximation errors are shown for the series expansion of the squared exponential of chordal distance for both varying number of truncation terms ρ and varying hyperparameters ℓ.

derivatives w.r.t. frequency ω_* have closed-form expressions given by

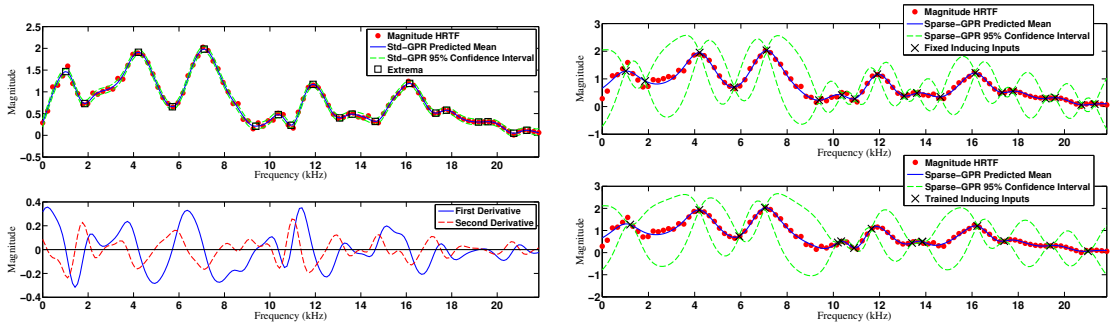
$$\frac{\partial \bar{f}_*}{\partial \omega_*} = \left[\frac{\partial K_{1*}}{\partial \omega_*}, \ldots, \frac{\partial K_{N*}}{\partial \omega_*}\right] \hat{K}^{-1} y, \quad \frac{\partial^2 \bar{f}_*}{\partial \omega_*^2} = \left[\frac{\partial^2 K_{1*}}{\partial \omega_*^2}, \ldots, \frac{\partial^2 K_{N*}}{\partial \omega_*^2}\right] \hat{K}^{-1} y,$$
$$\frac{\partial K_{i*}}{\partial \omega_*} = \frac{-2\alpha^2(\omega_* - \omega_i)}{(\lambda^2 + (\omega_* - \omega_i)^2)^2} e^{-C_{h_{i*}}/\ell^2}, \quad \frac{\partial^2 K_{i*}}{\partial \omega_*^2} = \frac{-2\alpha^2(\lambda^2 - 3(\omega_* - \omega_i)^2)}{(\lambda^2 + (\omega_* - \omega_i)^2)^3} e^{-C_{h_{i*}}/\ell^2}. \tag{3.42}$$

To find the zero-crossings, we use the partial derivatives of Eq. 3.42 and a standard iterative method (*Newton-Raphson*):

$$\text{Updtae: } \omega_{n+1} = \omega_n - \frac{\partial \bar{f}_{\omega_n}}{\partial \omega_n} \Big/ \frac{\partial^2 \bar{f}_{\omega_n}}{\partial \omega_n^2}, \quad \text{Terminate: } |\omega_{n+1} - \omega_n| < \tau. \tag{3.43}$$

Once the extrema are extracted, they can be further classified as either spectral notches or peaks by evaluating the sign of their second-order derivatives.

In small-scale experiments, we trained grid and sparse-grid GPs for 50 iterations on a sample HRTF (CIPIC subject 3, right ear, direction $(\theta, phi) = (2.3562, 0)$) and found the spectral extrema; The Newton-Raphson method converges in a several (< 10) iterations using a termination threshold of 10^{-5}. The initial inputs (frequency ω_0) are uniformly spaced in the frequency domain. Figure 3.10(a) shows the most prominent peaks and notches that are found.

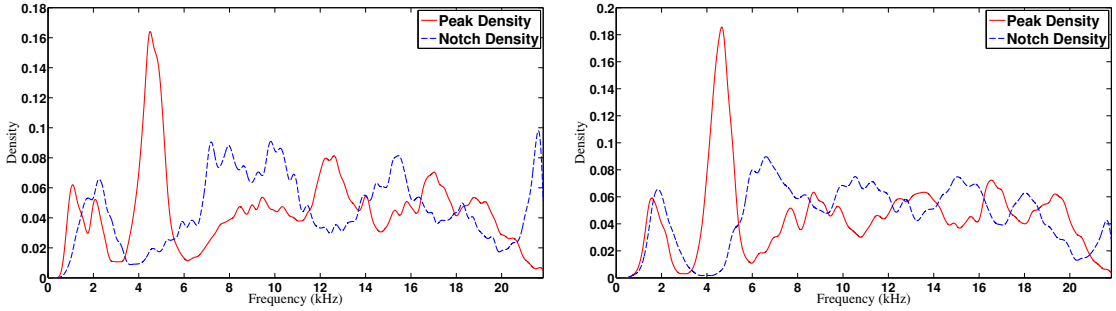


(a) Standard GPR (b) Sparse GPR

Figure 3.10: GP and sparse-GP posterior distributions for the magnitude HRTF responses are shown. Spectral extrema are extracted from the zero-crossing of the posterior mean's gradient.

In large-scale experiments, the spectra extrema are extracted for a large collection of HRTFs (all 45 CIPIC subject, right-ear, horizontal and median plane directions). The locations (frequencies) of the spectral notches and peaks are modeled by kernel density estimations (KDEs) (Gaussian kernels and optimized bandwidth [101]); see Figure 3.11. The horizontal plane extrema have a bi-modal distribution but do not exhibit a correspondence between notches and peaks; notch densities (frequencies $7 - 11$ and $14 - 16$ kHz) do not correspond with peak densities along the same frequencies. The median plane ex-

trema have a quad-modal distribution and exhibit a correspondence between notches and peaks; notch densities (centered along frequencies $6, 11, 15, 18$ kHz) have analogous peak densities shifted by $+2$ kHz. Both distributions are similar at the lower frequency ranges, which can be attributed to the initial torso and head reflections.



(a) KDE of Horizontal Plane Extrema

(b) KDE of Median Plane Extrema

Figure 3.11: Kernel density estimation (Gaussian) of pooled spectral extrema for GPs trained on horizontal and median plane HRTFs across all subjects, right ears.

3.7.3 Greedy Backward Subset Selection for Time-Delay Supports

The time-delay between acoustic wave-fronts that would reach a listener's left and right ears is an important spatial cue for sound-source localization. These cues, known as ITD, can be computed from the difference between the onset reflections of same-direction left and right ear HRIRs [102]. While ITDs can be derived from the spherical coordinate domain under simplified assumptions (spherical head [2], ellipsoid head [103]), the actual ITDs may deviate from these approximations due to slight asymmetries in shape of the head and relative positions of the ears. Thus, non-parametric methods such as GPs may be more accurate in modeling the ITDs.

We specify a GP using spherical input coordinates $X = X_s$ and ITD output observations y (CIPIC subject 3); a smooth squared exponential covariance function ($K(\theta, \phi) =$

$\alpha^2 e^{-\frac{C_h^2}{2\ell^2}}$) over the chordal distance C_h (Eq. 3.35) with hyperparameters (optimized via Eq. 3.6 for 50 iterations) is used. Moreover, the measurements that deviate from the GP prior assumptions, namely those that represent the asymmetries of the head, can be found via GBSS (Algorithm 4). Figure 3.12 shows two GP posterior ITD distributions (over the spherical coordinate domain) that are evidenced on the full set of ITD measurements and the GBSS ITD measurements. The subset-selected inputs reveal a slight bias/asymmetry towards the right hemisphere of the head.

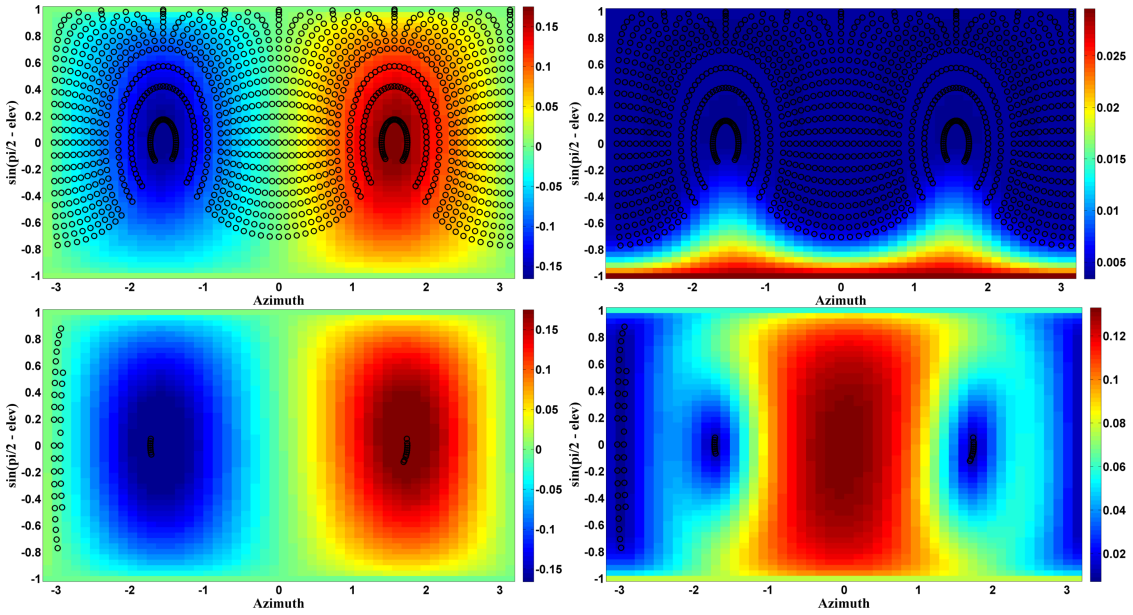


Figure 3.12: GP predicted ITD means are shown in the left-column; GP predicted ITD variances (95% confidence) are shown in the right-column. The measurement directions are marked $\circ$. Predictions evidenced on the full and GBSS ITD measurements belong to the top and bottom-rows respectively.

The quality of GPs evidenced on the GBSS ITD subsets can be evaluated via standard error metrics such as RMSE. Figure 3.13 shows the trade-off between the size of the remaining subset, the data LMH, and the RMSE. To bound the two learning curves, the opposite selection strategy (maximizing the remaining data LMH) is also implemented

which causes the remaining subset to withhold redundant measurements. The maximum gap between the best and worst case RMSEs and remaining LMH curves occurs at 50 remaining samples which indicates the number of supports (measurement direction and ITD) that characterizes all ITDs over the dataset/sphere.

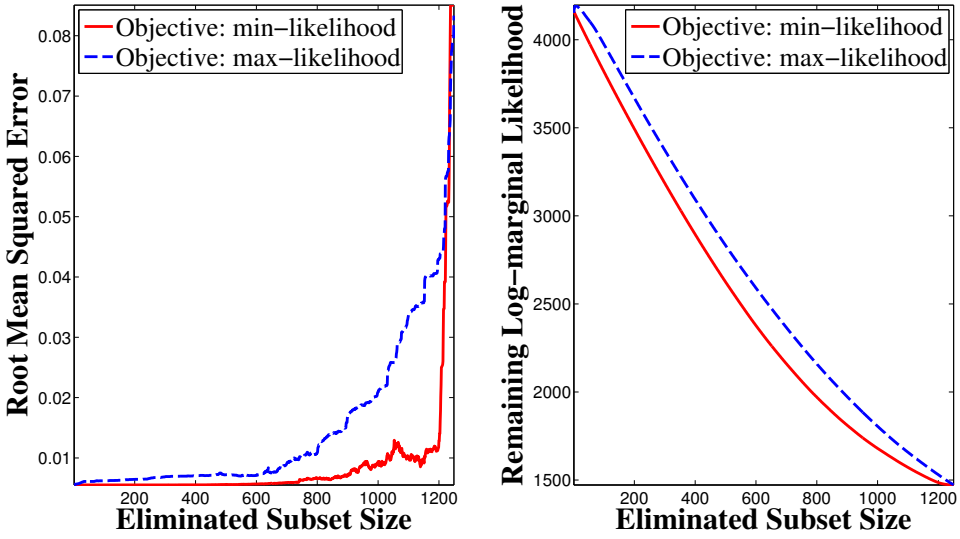


Figure 3.13: Learning curves are shown for RMSE prediction errors (left plot) and remaining data LMH (right) for GPs evidenced on the GBSS remaining samples as inputs are removed.

3.7.4 Greedy Backward Subset Selection for HRTFs

The search for representative data features serves an important function for reducing model-order and computational costs. For HRTF datasets, the subset-selection of magnitude responses over the spatial-frequency domains (under grid GP assumptions) would characterize the complexity/model-order of a description of the underlying sound field. Thus, analogous subset-selection experiments to section 3.7.3 are conducted for grid GPs specified on spatial-frequency inputs $X = X_s \times X_\omega$ and magnitude HRTF response out-

puts y (see section 3.7.2). GBSS, using grid GP's LMH as its objective function, ranks the inputs as either salient or redundant w.r.t. the trained GP priors.

In the small-case, the dataset is restricted to the subset of magnitude responses belonging to the inputs along the horizontal plane and the $9-13$ kHz frequency band for subject 12. Figure 3.14 shows grid GP's predicted magnitude response means for a variable R number of subset sizes; both selection strategies (minimize or maximize the remaining data LMH) are tested. The results show that 750 of the 1000 inputs can be removed (classified as redundant) before the mean reconstruction error (by grid GPs evidenced on the remaining dataset) at the missing inputs exceeds 5 dB.

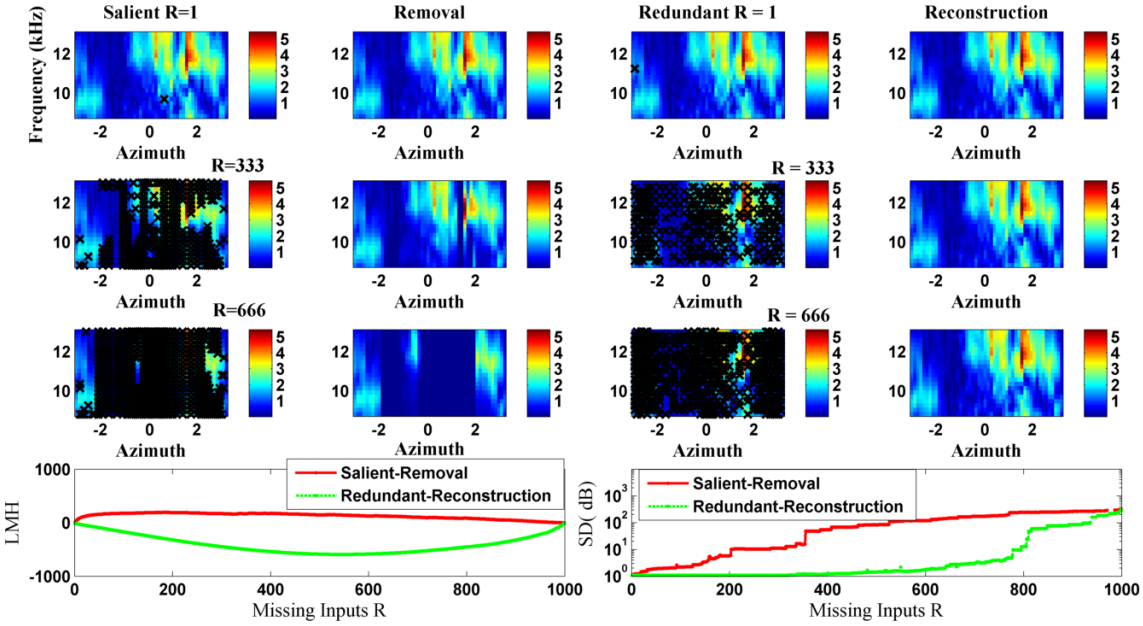


Figure 3.14: Grid GP's posterior magnitude response means are shown for inputs (azimuth plane and $8.8-13$ kHz inputs). The models are evidenced on the remaining subset-selected inputs (eliminated inputs are marked **X**). "Salient/redundant" refer to selection-strategies that "minimize/maximize" the remaining data LMH respectively; plots labeled "removal" and "reconstruction" fill in the missing inputs via grid GP inference. Bottom-row plots show the trade-off between subset-sizes and SD (over all predictions). Low SD indicates small error.

In the large-case, the previous experiment is repeated across all 45 CIPIC subjects

at different frequency bands; only the averaged learning curves are shown in Figure 3.15. We remark on the inflection points along each of the learning curves: For subsets that minimize the remaining LMH, the LMH curves are upwards-concave as inputs are removed. Inflection points at $R = 620, 720$, and 780 number of missing inputs, for increasing frequency ranges, suggest that the remaining inputs form the relevant subsets; grid GP evidenced on this remaining subset is able to reconstruct the missing set with low error. This is correlated with the SD errors which remain negligible upto these inflection points.

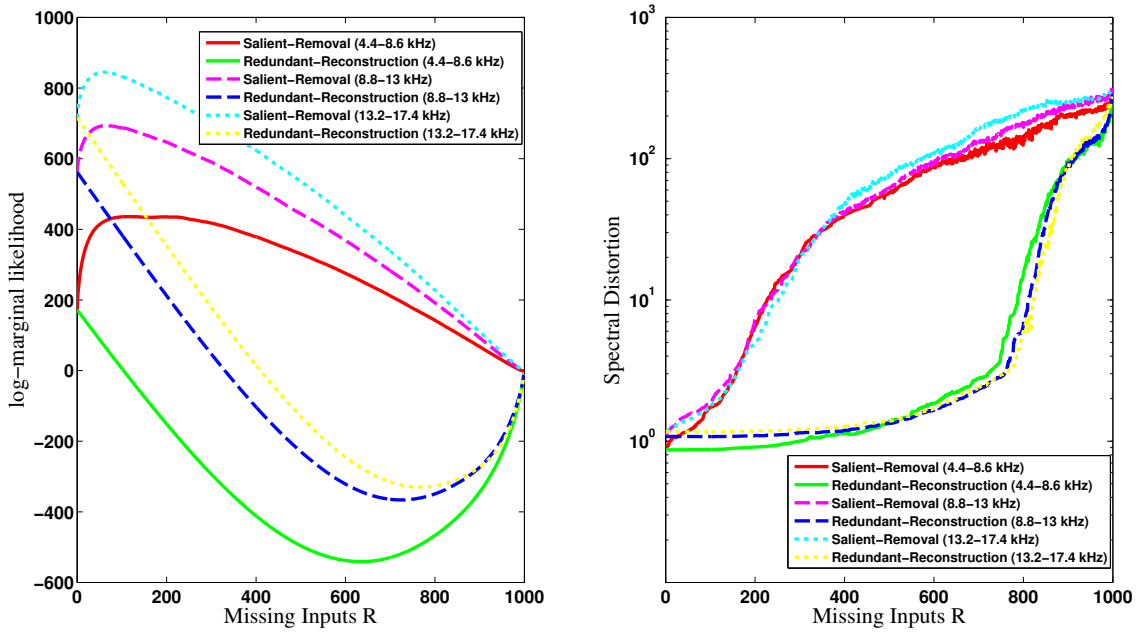


Figure 3.15: Average grid GP's (specified on the azimuth plane at different frequency ranges) remaining data LMH and SD (over all predictions) are shown for increasing subset-selected sizes. Low SD indicates small error.

The converse relation also holds for subsets chosen to maximize the remaining LMH; the LMH curves are downward-concave as more inputs are removed. Inflection points occur at $R = 200, 60$ and 50 number of missing inputs for increasing frequency ranges; subsequent inputs that are placed in the missing set after these inflection points decrease the remaining data LMH as GBSS begins to remove inputs that agree with the

GP priors. This is not indicated in the SD errors as the greatest rate of increase occurs between $R = 200$ to 300 where selecting against the GP zero-mean prior tends to increase the SD if low-magnitude observations are removed.

3.8 Conclusions

We have presented an overview of multidimensional grid GPs and its theoretical computational costs/savings from the use of efficient Kronecker product formulations. A connection between grid GP and Kronecker structures in GPLVM was remarked. Input measurement grid and TPK conditions were extended to sparse GP methods. Two problems for handling missing and extra data were posed and efficient solutions were presented; the missing data case was extended to fast GBSS for ranking inputs according to grid GP's remaining data LMH. The savings were empirically verified on high-dimensional synthetic data for full, missing, and extra data problems. Last, we applied grid and sparse grid GPs methods to interpolation, subset-selection, and missing data reconstruction problems on real-world HRTF datasets.

3.9 Appendix

3.9.1 Kronecker Product Identities

Unitary and binary matrix operations for Kronecker products have structured forms:

$$
\begin{aligned}
(A \otimes B)^{-1} &= A^{-1} \otimes B^{-1} && \text{Inverse-product} \\
(A \otimes B)^{T} &= A^{T} \otimes B^{T} && \text{Transpose-product} \\
A \otimes (B + C) &= A \otimes B = A \otimes C && \text{Bilinearity} \\
(A \otimes B)(C \otimes D) &= AC \otimes BD && \text{Mixed-product} \\
|A \otimes B| &= |A|^{p}|B|^{n}, \quad A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{p \times p} && \text{Deteminant} \\
\mathbf{tr}\,(A \otimes B) &= \mathbf{tr}\,(A)\,\mathbf{tr}\,(B) && \text{Trace}
\end{aligned}
$$

Properties of vectorization $\mathbf{vec}(A)$ (stacking columns of matrix A):

$$
\begin{aligned}
\mathbf{vec}(ABC) &= (C^{T} \otimes A)\mathbf{vec}(B) && \text{Vectorization 1} \\
&= (I_n \otimes AB)\mathbf{vec}(C) \\
&= (C^{T}B^{T} \otimes I_k)\mathbf{vec}(A), \quad A \in \mathbb{R}^{k \times l}, B \in \mathbb{R}^{l \times m}, C \in \mathbb{R}^{m \times n} \\
\mathbf{vec}(AB) &= (I_m \otimes A)\mathbf{vec}(B) && \text{Vectorization 2} \\
&= (B^{T} \otimes I_k)\mathbf{vec}(A), \quad A \in \mathbb{R}^{k \times l}, B \in \mathbb{R}^{l \times m}
\end{aligned}
$$

3.9.2 Relation to GPLVM

The GPLVM's data LMH (Eq. 3.14) can be expressed as that of grid GP's LMH:

$$
\begin{aligned}
\log p(Y|X) &= -\frac{1}{2}\left(\tilde{d}\log|\hat{C}| + \text{tr}(Y^T\hat{C}^{-1}Y) + N\log(2\pi)\right), \\
\text{tr}(Y^T\hat{C}^{-1}Y) &= \mathbf{vec}(Y)^T\mathbf{vec}(\hat{C}^{-1}Y) \quad \text{by outer product} \\
&= y^T\mathbf{vec}(\hat{C}^{-1}YI_{\tilde{d}}) \\
&= y^T(I_{\tilde{d}}\otimes\hat{C}^{-1})\mathbf{vec}(y) \\
&= y^T(I_{\tilde{d}}\otimes\hat{C})^{-1}y \\
&= y^T(I_{\tilde{d}}\otimes C + \sigma^2 I_N)^{-1}y \quad \text{by bilinearity.}
\end{aligned}
$$

The observation matrix $Y \in \tilde{\mathbb{N}} \times \tilde{}$ in the trace term vectorizes into $y \in \mathbb{R}^{N=\tilde{N}\tilde{d}}$. The covariance matrix now consists of $\tilde{d}$ diagonal blocks of the original matrix C as expressed by the Kronecker product.

3.9.3 Economical DTC

The DTC economical Gram matrix Σ [58] is expanded into products of KTPs with diagonal scaling:

$$
\begin{aligned}
\Sigma &= (\sigma^{-2}K_{uf}K_{fu} + K_{uu})^{-1} \\
&= \sigma^2(K_{uf}K_{fu} + \sigma^2 UZ^{1/2}Z^{1/2}U^T)^{-1} \\
&= \sigma^2(UZ^{1/2}(Z^{-1/2}U^TK_{uf}K_{fu}UZ^{-1/2} + \sigma^2 I)Z^{1/2}U^T)^{-1} \\
&= \sigma^2\Omega(\bar{Z} + \sigma^2 I)^{-1}\Omega^T, \quad \Omega = UZ^{-1/2}\bar{U},
\end{aligned}
\tag{3.44}
$$

for eigendecompositions $Z^{-1/2}U^T K_{uf} K_{fu} U Z^{-1/2} = \otimes_{i=1}^{D} \bar{U}_i \bar{Z}_i \bar{U}_i^T$, $U = \otimes_{i=1}^{D} U_i$ and $Z = \otimes_{i=1}^{D} Z_i$. The data LMH and gradient computations [104] are rearranged for KTVP operations in terms of matrix $\Psi = \sigma^{-2}\Sigma$ and vector $t = \Psi K_{uf} y$. The negative log-marginal likelihood and related terms are

$$
\begin{aligned}
-\log q(y|X) &= \frac{1}{2}\left(\log|\hat{Q}| + y^T \hat{Q}^{-1} y + N\log(2\pi)\right), \\
\log|\hat{Q}| &= (N-M)\log(\sigma^2) - \log|\Psi|, \quad |\Psi| = |Z^{-1}||(\bar{Z} + \sigma^2 I)^{-1}|, \\
y^T \hat{Q}^{-1} y &= \sigma^{-2} y^T (y - K_{fu}\Psi K_{uf} y) = \sigma^{-2} y^T (y - K_{fu} t).
\end{aligned}
\tag{3.45}
$$

The negative LMH gradient and related terms for Eqs. 3.19 and 3.15 are given by

$$
\begin{aligned}
-\frac{\partial \log q(y|X)}{\partial \Theta_i} &= \frac{1}{2}\left(\mathbf{tr}\left(\hat{Q}^{-1}\frac{\partial \hat{Q}}{\partial \Theta_i}\right) + \frac{\partial y^T \hat{Q}^{-1} y}{\partial \Theta_i}\right), \\
\mathbf{tr}\left(\hat{Q}^{-1}\frac{\partial \hat{Q}}{\partial \Theta_i}\right) &= \mathbf{tr}\left(2\frac{\partial K_{uf}}{\partial \Theta_i} K_{fu}\Psi\right) - \mathbf{tr}\left(\frac{\partial K_{uu}}{\partial \Theta_i} K_{uu}^{-1} K_{uf} K_{fu}\Psi\right), \\
&= 2\mathbf{diag}\left((\bar{Z} + \sigma^2 I)^{-1}\right)^T \mathbf{diag}\left(\Omega^T \frac{\partial K_{uf}}{\partial \Theta_i} K_{fu}\Omega\right) \\
&\quad - \mathbf{diag}\left((\bar{Z} + \sigma^2 I)^{-1}\right)^T \mathbf{diag}\left(\Omega^T \frac{\partial K_{uu}}{\partial \Theta_i} K_{uu}^{-1} K_{uf} K_{fu}\Omega\right), \\
\frac{\partial y^T \hat{Q}^{-1} y}{\partial \Theta_i} &= y^T K_{fu}\Psi \frac{\partial K_{uu}}{\partial \Theta_i}\Psi^T K_{uf} y - 2\sigma^{-2} y^T (I - K_{fu}\Psi K_{uf})\frac{\partial K_{fu}}{\partial \Theta_i}\Psi^T K_{uf} y \\
&= t^T \frac{\partial K_{uu}}{\partial \Theta_i} t - 2\sigma^{-2}(y^T \frac{\partial K_{fu}}{\partial \Theta_i} t - t^T K_{uf}\frac{\partial K_{fu}}{\partial \Theta_i} t).
\end{aligned}
\tag{3.46}
$$

For the trace term, the diagonals are efficiently computed over the products of KTPs with diagonal scaling. Only the partial derivative matrix $K_{uf}^{(l)}$ containing hyperparameter $\Theta_i \in K_l(x_j, x_k)$ is updated; the other blocks in the products of KTPs $K_{fu}K_{uf}\Omega$ are fixed. The cost of the diagonalizations is $\mathrm{O}\left(M + \sum_{i=1}^{D} m_i^{\{u\}^2}(m_i^{\{u\}} + m_i)\right)$ operations. For

the derivative term, the sparsity ratio $\rho_j = m_j^{\{u\}}/m_j$ per dimension factors into the cost of the rectangular KTVPs given by O $\left(N \sum_{i=1}^{D} m_i^{\{u\}} \prod_{j=i+1}^{D} \rho_j\right)$ operations.

3.9.4 Missing Data DTC

For efficient handling of missing data[6] by sparse-grid GPR, one can substitute low-rank downdates $K_{uf}K_{fu} - K_{ur}K_{ru} \rightarrow K_{uf}K_{fu}$ for terms appearing in matrix Σ and in the LMH (Eqs. 3.44, 3.45 and 3.46); matrix K_{ur} contains the r^{th} columns of matrix K_{uf} and zero-columns elsewhere. The rank-downdated economical Gram matrix $\hat{\Sigma} = \sigma^2(K_{uf}K_{fu} - K_{ur}K_{ru} + \sigma^2 K_{uu})^{-1}$ can be expressed in the form of Eq. 3.23 by analogous substitutions to Eq. 3.17 for matrix $(K_{uf}K_{fu} + \sigma^2 K_{uu})^{-1} \rightarrow \Sigma$ and $K_{ur} \rightarrow B \in \mathbb{R}^{M \times R}$ given by

$$\hat{\Sigma} = \sigma^2((K_{uf}K_{fu} + \sigma^2 K_{uu}) - K_{ur}K_{ru})^{-1} = \sigma^2(\Omega(\bar{Z} + \sigma^2 I)^{-1}\Omega^T + B^{(R)}DB^{(R)^T}).$$

The missing data entries are handled in the subspace spanned by the inducing inputs **u**; the costs of computing matrices $B^{(R)}$ and D via Eq. 3.23 are O $\left(R^2M + RM \sum_{i=1}^{D} m_i^{\{u\}}\right)$ operations and O (RM) space.

3.9.5 Index Operations

The conversions between the general column index q to its D-KTP column index $\bar{q}$ are used for VKTP operations to generate the column update/downdate vectors in the missing data problem.

[6] Handling extra data entries is analogous to the missing data case with low-rank updates instead

Algorithm 7 General column index q to D-KTP column index $\bar{q}$ (**ColToK**)

Require: Column sizes $\ddot{m}_i$ for Kronecker factors $[C_1 \in \mathbb{R}^{m_1 \times \ddot{m}_1}, \dots C_D \in \mathbb{R}^{m_D \times \ddot{m}_D}]$, general column index q

1: **for** $i = D$ to 1 **do**
2: $\quad \bar{q}_i \leftarrow ((q-1) \mod \ddot{m}_i) + 1$
3: $\quad q \leftarrow \text{ceil}(q/\ddot{m}_i)$
4: **end for**
5: **return** $\bar{q} \in \mathbb{N}^D$

Algorithm 8 D-KTP column index $\bar{q}$ to general column index q (**KToCol**)

Require: Column sizes $\ddot{m}_i$ for Kronecker factors $[C_1 \in \mathbb{R}^{m_1 \times \ddot{m}_1}, \dots C_D \in \mathbb{R}^{m_D \times \ddot{m}_D}]$, D-KTP column index $\bar{q}$

1: $N \leftarrow \prod_{i=1}^{D} \ddot{m}_i$
2: $s \leftarrow \text{cumprod}(\ddot{m})$ $\quad \backslash\backslash$ D-length cumulative product, $s_j = \prod_{i=1}^{j} \ddot{m}_i$
3: $s \leftarrow N/s$
4: $q \leftarrow 1 + \sum_{i=1}^{D} (\bar{q}_i - 1) s_i$
5: **return** $q \in \mathbb{N}$

3.9.6 Spherical Covariance Function Representations

A real continuous function $K(\gamma)$ is said to be a valid covariance function on the sphere [96] if and only if it can be expressed as follows:

$$K(\gamma) = \sum_{n=0}^{\infty} b_n P_n(\cos\gamma), \quad b_n \geq 0, \quad \sum_{n=0}^{\infty} b_n < \infty, \quad \gamma \in [0, \pi], \tag{3.47}$$

where $P_n(\cos\gamma)$ are the Legendre polynomials, γ the central angle between (θ_i, ϕ_i), (θ_j, ϕ_j), and b_n depends on the choice of the covariance function [95]. The Legendre addition theorem is given by

$$P_n(\cos\gamma) = \frac{4\pi}{2n+1} \sum_{m=-n}^{n} Y_n^m(\theta_i, \phi_i) \bar{Y}_n^m(\theta_j, \phi_j), \tag{3.48}$$

which when combined with Eq. 3.47 gives Eq. 3.36.

Chapter 4: Heterogeneous HRTF Dataset Fusion via Gaussian Processes

4.1 Introduction

Head-Related Transfer Function (HRTF) measurement and extraction are important tasks for personalized-spatial audio. 3D audio synthesis is based on the human ability to localize sound using monaural and binaural cues of how a sound-source's acoustic wave scatters off of the listener's anatomy (torso, head, and outer ears). The ratio of the Fourier Transform of this wave, measured at the listener's eardrum to that which would have been present at the head-center location in the absence of the listener, is called the HRTF [13]. While many research labs have their own apparatuses for measuring HRTFs for human listeners, very few comparisons have been made between the data measurements collected over common subjects. In theory, such a comparison is unnecessary as ideal HRTFs would be recorded in a free-field and should not contain the effects of the environment. In practice, many distortions between HRTF datasets over common subjects can be plainly observed and which are the cause of a significant amount of inter-lab variance.

To address this problem, a large round-robin activity was organized [105] where HRTFs are collected over a Neumann KU-100 dummy head by different labs. The collection, referred to as the "Club Fritz" database, contains the mannequin's HRTF mea-

surements[1] from 7 different labs. Moreover, each lab used their own measurement apparatuses, which resulted in 7 distinct measurement grids over the spherical coordinate domain. Fig. 4.1 shows the HRTF measurement grids used by each lab which all vary substantially over the sphere and thereby making any one-to-one correspondences between the HRTFs along the same measurement directions difficult.

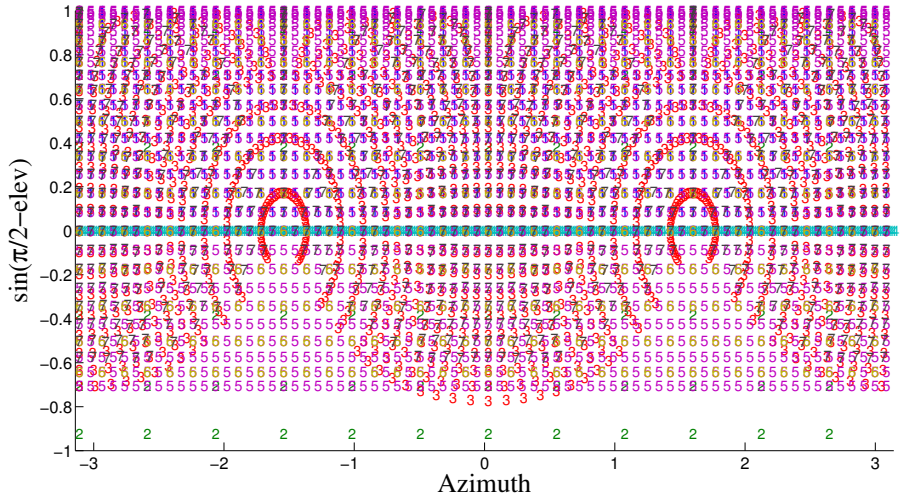


Figure 4.1: Mercator projection of measurement grids are shown for "Club Fritz" Neumann HRTFs. For anonymity, the source institutions are indicated by the lab numbers.

We propose a Bayesian data-fusion method, based on Gaussian process (GP) regression (GPR) [46], to model the underlying sound-field from which all common-subject HRTFs are drawn from. Formally, a GP is random field $f(x)$ where any finite subset of its random variables $f = \{f(x_1), \ldots, f(x_N)\}$ (indexed at inputs x) are jointly Gaussian and thus defined by prior mean and covariance functions. The GP priors (mean and covariance functions) are responsible for the distribution of the realizations of $f(x)$ in the absence of observations; the mean function can be set to 0 without loss of generality.

[1] HRTF measurements are preprocessed by recovering their minimum-phase Head-Related Impulse Responses (HRIR) to remove time-delay, resampling the HRIR to 44100 kHz, taking the magnitude of the discrete Fourier transform of the first 256 taps, truncating to the $0-18$ kHz range, and scaling the magnitude range to $(0, 1)$. We use the term *HRTF measurement* to refer exclusively to HRTF magnitude.

The covariance function asserts that $f(X)$, at topologically similar indexes X, produces similar realizations with high probability. In the presence of observations (output) y at indexes X, the random field $f(X_*)$ (test inputs X_*) conditioned on $f(X) = y$ is also jointly normal and specified by so-called "posterior" mean and covariance functions (see section 4.2 for complete derivation). This gives a probablistic description of the output domain where observations y can be evaluated in terms of likelihoods of having been drawn from either prior or posterior distributions.

To apply the GP framework to HRTF measurements, we model the subject's sound-field magnitude responses (realizations of $f(x)$) as a collection of random variables indexed by spherical coordinate (azimuth and elevation) and frequency (wave number) tuples $x = (\theta, \phi, \omega)$. A separable and stationary covariance function is specified over the spatial-frequency input domains, which coincides with the observation that magnitude HRTFs are often smooth in both spherical coordinate and frequency domains. For known HRTF measurements $\mathcal{D}_i = \left(X^{\{i\}}, y^{\{i\}}\right)$ (dataset i), the sound-field $f(X_*)|f(X^{\{i\}}) = y^{\{i\}}, X^{\{i\}})$ at any test X_* (directions and frequencies) is characterized by a posterior normal distribution. Thus, realizations of the sound-field (conditioned on an HRTF dataset of N observations) are simply drawn from a N-dimensional joint normal distribution). This formulation is based on previous works of so-called "grid GP" models [17, 81] and is equivalent to GP based HRTF interpolation [16, 18].

4.1.1 Problem Formulation

While specifying a sound-field by a GP conditioned on a single HRTF dataset is feasible, the likelihoods of sampling the measurements of other datasets from the sound-field is low. This is due to large inter-lab variances between HRTF measurements at nearby or identical directions; such variances may have numerous origins from measurement noise, positioning errors, non-omnidirectional directivity patterns, temperature dependent equipment transfer function drifts, from incompatible free-field equalizations etc. The problem of data-fusion can thus be formulated as learning a set of transformations (representing one or more of the above origins) for each dataset that brings it closer to a reference sound-field. The reference sound-field has numerous instantiations such as GPs conditioned on the elements of the powerset of datasets $\{\mathcal{D}_1 \ldots, \mathcal{D}_7\}$; two notable cases are the sound-fields belonging to individual datasets and the averaged sound-field generated from the HRTF superset. Furthermore, learning the transformations can follow several optimization techniques (e.g. maximum likelihood of sampling from the posterior reference sound-field, maximum log-marginal likelihood (LMH) (see section 4.2) of sampling both reference and transformed datasets from a common prior reference sound-field). This work uses individual dataset sound-fields and the LMH objective function to optimize transformations.

Two transformations belonging to the category of "incompatible free-field equalizations" are learned: The first transform is frequency-domain equalization where all HRTFs are multiplied (point-wise) by a filter (see section 4.3.1). Equalization filters are commonly applied to source-signals to either suppress a range of frequencies or to add ad-

ditional gain; some labs may have used this technique to remove low frequencies (the effects of torso and shoulder reflections) and to compensate (remove) the measurement apparatuss' transfer functions. The second transform is time-domain windowing, which is equivalent to the convolution operation with a filter performed in the frequency-domain (treated as if time-domain) (see section 4.3.2). This technique is commonly used to remove the effects of late reflections that would have been caused by sound scattering off of distant objects such as ground/walls. In both cases, the filter coefficients belonging to each dataset w.r.t. the reference dataset are jointly learned. Moreover, it is shown that these two transformations generalize most of the variances between inter-lab measurement processes; experiments show that sound-fields specified over the transformed datasets and similar to the that of the reference datasets compared to non-transformed cases (see section 4.4).

4.2 Gaussian Process Regression

In a general regression problem, one predicts a scalar target variable y from a D-dimensional vector x of independent variables based on a collection of available observations (measurements). In a parametric model, the problem is one of estimating model parameters based on the data. When a parametric model is unknown, a common Bayesian approach of inference assumes that observations y are generated by an unknown (latent) function $f(x)$ and is corrupted by additive (Gaussian) noise $y = f(x) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma^2)$ (noise term ϵ is zero centered with constant variance σ^2).

For a GP f, the latent function is modeled as random variables where any finite

collection $f = [f(x_1), f(x_2), \ldots, f(x_N)]$, indexed at $X = [x_1, \ldots, x_N]$, has a joint N-dimensional normal distribution that is specified by the prior mean function $m(x)$ and covariance function $K(x_i, x_j)$. The prior mean $m(x)$ can be specified as 0 without loss of generality. The covariance function generates a covariance (Gram) matrix $K_{ff} \in \mathbb{R}^{N \times N}$, representing the pair-wise covariance function evaluations between inputs in X. For N number of random variables f at known inputs X and N_* number of random variables $f_* = f(X_*)$ at test inputs X_*, the joint-prior distribution is given by

$$\begin{bmatrix} f + \epsilon \\ f_* \end{bmatrix} \sim \mathcal{N}\left(m(x), \begin{bmatrix} K(X,X) + \sigma^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix} \right), \tag{4.1}$$

for matrices $K_{f*} = K(X, X_*) \in \mathbb{R}^{N \times N_*}$, and $K_{**} = K(X_*, X_*) \in \mathbb{R}^{N_* \times N_*}$. GP inference simply conditions the random variables f_* on $f + \epsilon = y$ (Eq. 4.1), which has a N_*-dimensional posterior joint normal distribution given by

$$\begin{aligned} f_* | X, y, X_* &\sim \mathcal{N}(\bar{f}_*, \mathbf{cov}\,(f_*)), \\ \bar{f}_* = E[f_* | X, y, X_*] &= K_{f*}^T \hat{K}^{-1} y, \quad \mathbf{cov}\,(f_*) = K_{**} - K_{f*}^T \hat{K}^{-1} K_{f*}. \end{aligned} \tag{4.2}$$

Thus, GPs provides a probabilistic description of f over entire input domain and is able to report the expected means (posterior mean vector $\bar{f}_*$) and the confidence (posterior covariance matrix $\mathbf{cov}\,(f_*)$) at X_* in the presence of data.

4.2.1 Spatial-Frequency Covariance Functions for Sound-Fields

The choice of the prior covariance function K determines the "smoothness/correlatedness" of latent function realizations $f(X)$ at nearby X. The goodness-of-fit of observations y w.r.t. the GP prior assumptions can be evaluated by marginalizing the data likelihoods (y from $f(x)+\epsilon$) and priors (realizations of $f(x)$ drawn from the GP prior distribution) over all possible realizations of $f(x)$; this quantity is the so-called data LMH and obtains an analytic form that is useful for evaluating the selection of covariance functions. Moreover, covariance functions can be further characterized by their hyperparameters (Θ_i) and optimized by maximizing the data LMH via hill-climbing methods such as steepest ascent. Both the LMH and its partial derivative w.r.t. Θ_i are given by

$$\begin{aligned} \log p(y|X) &= -\frac{1}{2}\left(\log|\hat{K}| + y^T\hat{K}^{-1}y + N\log(2\pi)\right), \\ \frac{\partial \log p(y|X)}{\partial \Theta_i} &= -\frac{1}{2}\left(\mathbf{tr}\left(\hat{K}^{-1}P\right) - y^T\hat{K}^{-1}P\hat{K}^{-1}y\right), \end{aligned} \tag{4.3}$$

where matrix $P = \partial\hat{K}/\partial\Theta_i$.

For sound-fields characterized by the GP magnitude frequency responses f_* at $x_* = (\theta_*, \phi_*, \omega_*)$, it is possible to specify the covariance function as a product of separable (functions restricted to different domains) covariance functions on spherical-coordinate and frequency domains [16, 18]. Moreover, HRTF inputs have the unique parameterization given by the Cartesian outer-product $X = X^{(\theta\phi)} \times X^{(\omega)}$. This allows the Gram

matrix K_{ff} to be expressed by so-called Kronecker tensor products (KTP) [57] given by

$$K_{ff} = K_1(X^{(\theta\phi)}, X^{(\theta\phi)}) \otimes K_2(X^{(\omega)}, X^{(\omega)}), \tag{4.4}$$

between covariance evaluations restricted to inputs in $X^{(\theta\phi)}$ and $X^{(\omega)}$ respectively. Efficient KTP matrix algorithms for GP inference and hyperparameter training can also be found in [17, 81].

We adopt the stationary covariance function $K(C_h, r) = K_1(C_h)K_2(r)$ of the product of the Matérn ($\nu = 3/2$) covariance functions [46] for Chordal distance C_h and the spectral density of the *Ornstein-Uhlenbeck* (OU) auto-covariance [94] for frequency distance $r = |\omega_i - \omega_j|$ given by

$$\begin{aligned} K_1(C_h) = \left(1 + \frac{\sqrt{3}C_h}{\ell}\right) e^{-\frac{\sqrt{3}C_h}{\ell}}, \quad K_2(r) = \frac{2\alpha^2\lambda^2}{\lambda^2 + r^2}, \\ C_h = 2\sqrt{\sin^2\left(\frac{\theta_j - \theta_i}{2}\right) + \sin\theta_i \sin\theta_j \sin^2\left(\frac{\phi_i - \phi_j}{2}\right)}. \end{aligned} \tag{4.5}$$

Hyperparameters α, λ, and ℓ are the global-scale factor, the rate of mean drift to 0 in the OU process, and the characteristic *length-scales*[2] respectively. Other combinations of covariance products including Matérn $\nu = \{1/2, 5/2, \infty\}$ lead to lower data-LMH estimates in Eq. 4.3 by individual datasets $\mathcal{D}_i$ after hyperparameter training.

[2]Zero-crossings of 1D functions drawn from the GP prior with mean 0

4.3 Data Fusion and Transformations

We first establish notation as follows: For $T = 7$ number of HRTF datasets, let inputs $X = \left\{X^{\{1\}}, \ldots, X^{\{T\}}\right\}$ correspond to observations $y = \left[y^{\{1\}}; \ldots; y^{\{T\}}\right]$. Let function $g_t(y)$ with parameters $\Theta^{\{t\}}$ transform all but the reference dataset ($y^{\{i\}} \forall i \neq T$ s.t. $y^{\{t\}}$ remains constant); $\Theta^{\{t\}}$ contains separate filter coefficients for each dataset $\mathcal{D}_{i \neq t}$.

The sound-field is specified as follows: A reference GP is initially specified on only dataset $\mathcal{D}_t$ and its covariance function is trained (hyperparameters $\Theta_i^{\{K,t\}}$ are optimized via Eq. 4.3). A second GP, representing the fused sound-field, is specified on the non-transformed data (X, y) with the reference GP's covariance function (transformation parameters produce the identity operation). The transformation parameters are optimized by maximizing the sound-field GP's LMH (via partial derivative) given by

$$\mathcal{L}_t = -\frac{1}{2}\left(\log|\hat{K}| + g_t(y)^T\gamma + N\log(2\pi)\right), \quad \gamma = \hat{K}^{-1}g_t(y), \quad \frac{\partial\mathcal{L}_t}{\partial\Theta_i^{\{t\}}} = -\gamma^T\frac{\partial g_t(y)}{\partial\Theta_i^{\{t\}}}. \tag{4.6}$$

After the transformation parameters are trained, the fused sound-field is thus given by the GP conditioned on the transformed data $(X, g_t(y))$; if the transformations are able to model the inter-data variances, then the GP posterior distribution will be similar to that of the reference GP. The full process is shown in Fig. 4.2. The two types of transformations are described below.

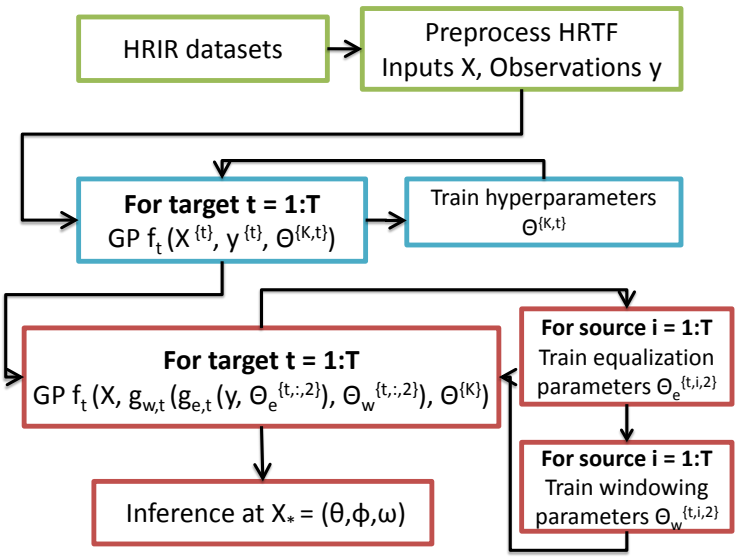


Figure 4.2: Reference GPs are trained and whose covariance function is reused for a second GP specified over the combined datasets. The transformation filter coefficients are trained and the fused sound-field is given by the second GP conditioned on the transformed datasets.

4.3.1 Equalization-Transform

The *equalization-transform* applies (diagonal-matrix vector product) a common separable filter to the measurements in the spatial-frequency domains given by

$$\begin{aligned} g_t(y) &= \mathbf{diag}\left[\Phi_t^{\{1\}}, \ldots, \Phi_t^{\{t-1\}}, 1^{N_t}, \Phi_t^{\{t+1\}}, \ldots, \Phi_t^{\{T\}}\right] y, \\ \Phi_t^{\{i\}} &= \left(\Theta^{\{t,i,1\}} \otimes \Theta^{\{t,i,2\}}\right) \in \mathbb{R}^{|X_{\theta\phi}^{\{i\}}||X_{\omega}^{\{i\}}|}. \end{aligned} \tag{4.7}$$

Constant filter coefficients, $1^{N_t} \in \mathbb{R}^{N_t}$, are set to the vector of ones as to perform the identity transform on the reference dataset $Y^{\{t\}}$. Variable filter coefficients $\Phi_t^{\{i\}}$ are Kronecker diagonal-products between filter the spatial filter coefficients $\Theta^{\{t,i,1\}}$ and the frequency filter coefficients $\Theta^{\{t,i,2\}}$. To optimize these coefficients, we can maximize the LMH via the partial derivatives of transform $g_t(y)$ w.r.t. the filter coefficients ($u = \partial g_t(y)/\partial \Theta_j^{\{t,i,1\}}$

and $v = \partial g_t(y)/\partial \Theta_j^{\{t,i,2\}}$) given by

$$
\begin{aligned}
u &= \mathbf{diag}\left[0^{N_1}, \ldots, 0^{N_{t-1}}, \frac{\partial \Phi_t^{\{i\}}}{\partial \Theta_j^{\{t,i,1\}}}, 0^{N_{t+1}}, \ldots, 0^{N_T}\right] y, \\
v &= \mathbf{diag}\left[0^{N_1}, \ldots, 0^{N_{t-1}}, \frac{\partial \Phi_t^{\{i\}}}{\partial \Theta_j^{\{t,i,2\}}}, 0^{N_{t+1}}, \ldots, 0^{N_T}\right] y, \\
\frac{\partial \Phi_t^{\{i\}}}{\partial \Theta_j^{\{t,i,1\}}} &= e_j \otimes \Theta^{\{t,i,2\}}, \quad \frac{\partial \Phi_t^{\{i\}}}{\partial \Theta_j^{\{t,i,2\}}} = \Theta^{\{t,i,1\}} \otimes e_j,
\end{aligned}
\tag{4.8}
$$

where e_i the i^{th} column of the identity matrix.

If the spatial filter coefficients ($\Theta^{\{t,i,1\}} = 1^{|X_{\theta\phi}^{\{i\}}|}$) are fixed, then optimizing for the frequency coefficients ($\Theta^{\{t,i,2\}}$) can be interpreted as equalizing all magnitude HRTFs in dataset $\mathcal{D}_i$ by a common filter. Conversely, fixing the frequency cofficients ($\Theta^{\{t,i,2\}} = 1^{|X_{\omega}^{\{i\}}|}$) and optimizing for spatial filter coefficients ($\Theta^{\{t,i,1\}}$) uniquely scales the full magnitude spectrum for each measurement direction in dataset $\mathcal{D}_i$.

Optimizing the filter coefficients is efficient as the LMH $\mathcal{L}_t$ is quadratic w.r.t. each $\Theta_j^{\{t,i,1\}}$ and $\Theta_j^{\{t,i,2\}}$. Setting the partial derivatives in Eq. 4.6 to zero, their solutions are given by

$$
\Theta_j^{\{t,i,1\}} = -\frac{g_t(y)_u^T \hat{K}^{-1} u}{u^T \hat{K}^{-1} u}, \quad \Theta_j^{\{t,i,2\}} = -\frac{g_t(y)_v^T \hat{K}^{-1} v}{v^T \hat{K}^{-1} v}, \tag{4.9}
$$

where $g_t(y)_u = g_t(y) - \Theta_j^{\{t,i,1\}} u$ and $g_t(y)_v = g_t(y) - \Theta_j^{\{t,i,2\}} v$. Thus, the filter coefficient parameters will quickly converge as the LMH $\mathcal{L}_t$ monotonically increases.

4.3.2 Window-Transform

The *window-transform* simulates time-domain windowing (point-wise product) by the equivalent convolution operation in the frequency domain. The convolution operation can me formulated as a symmetric Toeplitz-matrix vector product given by

$$\begin{aligned} g_t(y) &= \mathbf{bdg}\left[\Phi_t^{\{1\}}, \ldots, \Phi_t^{\{t-1\}}, I_{N_t}, \Phi_t^{\{t+1\}}, \ldots, \Phi_t^{\{T\}}\right] y, \\ \Phi_t^{\{i\}} &= \mathbf{Tps}\left(\Theta^{\{t,i,1\}}\right) \otimes \mathbf{Tps}\left(\Theta^{\{t,i,2\}}\right), \end{aligned} \tag{4.10}$$

where $\mathbf{bdg}\left[A_1, A_2\right]$ generates a block-diagonal matrix whose diagonal elements are square matrices A_1 and A_2 and the off-diagonal elements are 0's. The filter coefficients ($\Phi_t^{\{i\}}$) are given by Kronecker products of symmetric-Toeplitz matrices $\mathbf{Tps}\left(a\right)_{jk} = a_{|j-k|+1}$ generated from the spatial filter coefficients ($\Theta^{\{t,i,1\}}$) and the frequency filter coefficients ($\Theta^{\{t,i,2\}}$) (identical to Eq. 4.7).

Optimizing these filter cofficients (maximizing LMH) is efficient as the formulation is analogous to that of the equalization transform in section 4.3.1. The partial derivatives of the transformation w.r.t. the spatial and frequency filter cofficients ($u = \partial g_t(y)/\partial\Theta_j^{\{t,i,1\}}$ and $v = \partial g_t(y)/\partial\Theta_j^{\{t,i,2\}}$) are given by

$$\begin{aligned} u &= \mathbf{bdg}\left[0_{N_1}, \ldots, 0_{N_{t-1}}, \frac{\partial\Phi_t^{\{i\}}}{\partial\Theta_j^{\{t,i,1\}}}, 0_{N_{t+1}}, \ldots, 0_{N_T}\right] y, \\ v &= \mathbf{bdg}\left[0_{N_1}, \ldots, 0_{N_{t-1}}, \frac{\partial\Phi_t^{\{i\}}}{\partial\Theta_j^{\{t,i,2\}}}, 0_{N_{t+1}}, \ldots, 0_{N_T}\right] y, \\ \frac{\partial\Phi_t^{\{i\}}}{\partial\Theta_j^{t,i,1}} &= \mathbf{Tps}\left(e_j\right) \otimes \mathbf{Tps}\left(\Theta^{\{t,i,2\}}\right), \quad \frac{\partial\Phi_t^{\{i\}}}{\partial\Theta_j^{t,j,1}} = \mathbf{Tps}\left(\Theta^{\{t,i,1\}}\right) \otimes \mathbf{Tps}\left(e_j\right), \end{aligned} \tag{4.11}$$

where $0_{N_i} \in \mathbb{R}^{N_i \times N_i}$ is the zero-matrix. The local solutions for each filter coefficient has the closed-form expression identical to Eq. 4.9 after the appropriate substitutions.

4.3.3 Composition

It is possible to specify the a transformation function as the composition of the window-transform $g_{w,t}$ (Eq. 4.10) and the equalization-transform $g_{e,t}$ (Eq. 4.7) given by

$$g_t(y) = g_{w,t}(g_{e,t}(y)). \tag{4.12}$$

The filter coefficients can be optimized by modifying Eq. 4.9: For window coefficients $\Theta_w^{\{t\}} \in g_{w,t}$, observation vector y is replaced with $g_{e,t}(y)$ in Eqs. 4.10 and 4.11. For equalization filter coefficients $\Theta_e^{\{t\}} \in g_{e,t}$, both the partial derivatives $\partial\Phi_t^{\{i\}}/\partial\Theta_j^{t,i,1}$ and $\partial\Phi_t^{\{i\}}/\partial\Theta_j^{t,j,1}$ in Eq. 4.8 are left-multiplied by parameters $\Phi_t^{\{i\}}$ from Eq. 4.10.

4.4 Experiments

For computational costs, we abridge the HRTF measurements to those restricted to the horizontal and median planes. Reference sound-fields are specified for each dataset. The reference GP's covariance function hyperparameters are optimized for 100 iterations (Eq. 4.3). The compote transform (Eq. 4.12) is specified and its filter coefficients are initialized to perform the identity operation ($\Theta_e^{\{t,i,1\}} = 1^{|X_{\theta\phi^{\{i\}}}|}$, $\Theta_e^{\{t,i,2\}} = 1^{|X_{\omega^{\{i\}}}|}$, $\Theta_w^{\{t,i,1\}} = e_1$, and $\Theta_w^{\{t,i,2\}} = e_1$). Filter cofficients are then trained (Eq. 4.6) for 5 iterations for all source and reference datasets $\mathcal{D}_i$.

Figs. 4.3 and 4.4 show both the original datasets and fused sound-fields (GP posterior magnitude response means) along the horizontal and median plane directions. Large variances between the datasets are apparent; the presence of torso and shoulder reflections (low frequency response along plots row 1 columns $1, 4$) is present in two of the labs. High frequency responses are suppressed in two of the labs (row 1 columns $3, 6$ and row 4 columns $3, 7$). The fused sound-fields (rows $3, 6$), specified on the transformation datasets, are similar to the reference sound-fields (rows $1, 4$).

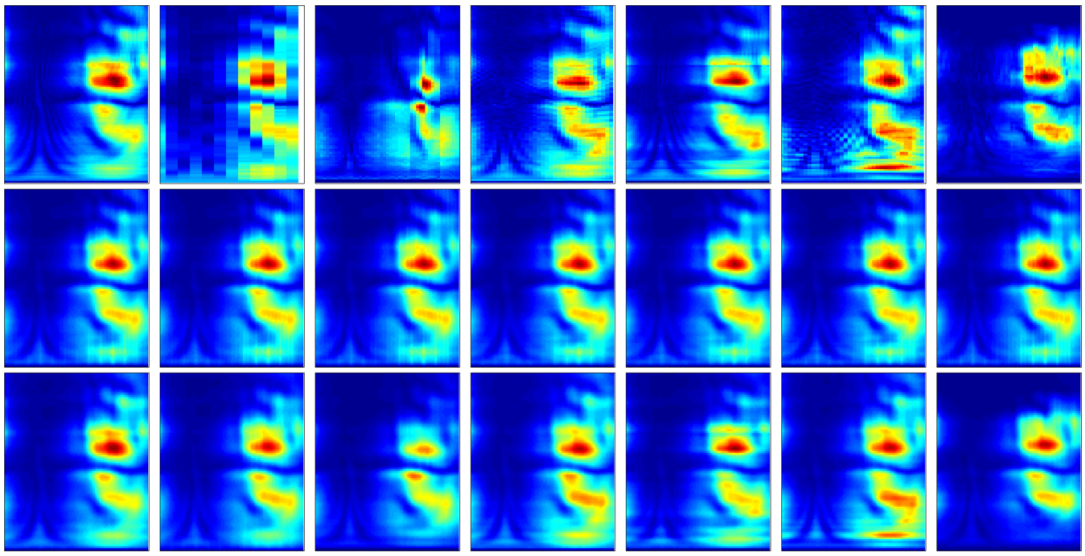


Figure 4.3: Plots in row 1 are the reference HRTFs (labs $1-7$) on horizontal plane (x-axis $-\pi < \phi < \pi$ and y-axis $0 < \omega < 18$ kHz). Plots along different columns refer to the reference dataset in row 1. Rows 2 and 3 are the sound-fields (GP predicted magnitude response means conditioned on non-transformed datasets and transformed datasets respectively).

The fused sound-fields can be evaluated against the reference sound-field by comparing their respective GP posterior magnitude response means ($\bar{f}_*$ in Eq. 3.5) evaluated at the reference inputs $X^{\{t\}}$. One metric is the signal-to-distortion ratio (SDR) given by

$$\text{SDR}_\omega = 10 \log_{10} \frac{\sum_{i=1}^{N_*} H_\omega(\theta_i, \phi_i)^2}{\sum_{i=1}^{N_*} (H_\omega(\theta_i, \phi_i) - \hat{H}_\omega(\theta_i, \phi_i))^2}, \tag{4.13}$$

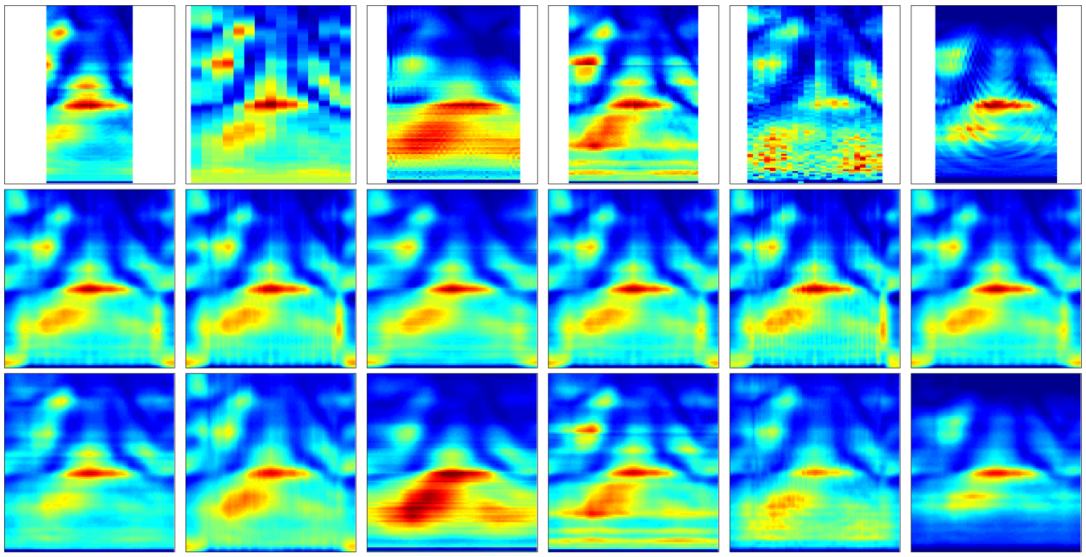


Figure 4.4: Plots in row 1 are the reference HRTFs (labs $1, 2, 3, 5, 6, 7$) on median plane (x-axis $-\pi < \theta < \pi$ and y-axis $0 < \omega < 18$ kHz). Plots along different columns refer to the reference dataset in row 1. Rows 2 and 3 are the sound-fields (GP predicted magnitude response means conditioned on non-transformed datasets and transformed datasets respectively).

where $H_\omega(\theta_i, \phi_i) = y_{\omega,\theta_i,\phi_i}$ is the reference magnitude responses and $\hat{H}_\omega(\theta_i, \phi_i) = \bar{f}_{\omega,\theta_i,\phi_i}$ is the predicted mean responses.

Figs. 4.5 and 4.6 show that the SDRs of the fused sound-field's after learning the transformations, are larger (lower error) than that of the non-transformed ones across most frequency bands for horizontal and median planes respectively. Larger SDR discrepancies between median-plane HRTFs[3] than that of the horizontal-plane for transformed and control cases may suggest greater measurement sensitivities along the former directions. Fused dataset target plots $\{6, 5\}$ and $\{6, 7\}$ have the highest SDRs relative to the control. The equalization weights for each frequency appear continuous in log-space. The window weights exhibit periodicity similar to window functions in the Fourier domain. Moreover, both the window and equalization weights learned for median and horizontal

[3]One horizontal-plane only dataset was omitted as a median-plane target.

plane HRTFs of same target $\mathcal{D}_t$ are similar and thus consistent over two different regions of the sphere.

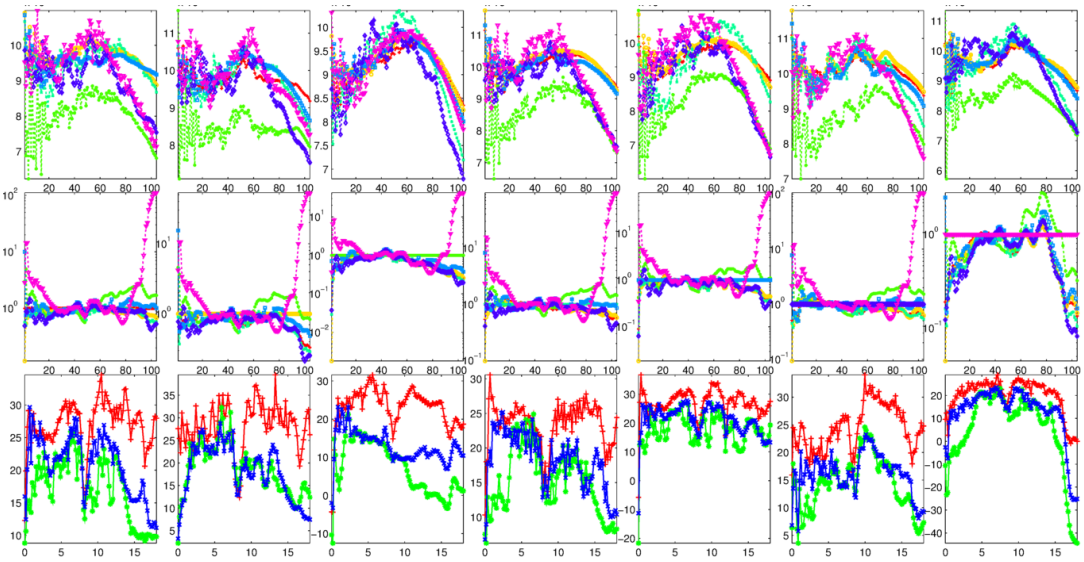


Figure 4.5: Rows 1 and 2 show the horizontal-plane-trained window-filter coefficients (after min-phase reconstruction into time-domain) and the equalization-transform coefficients (absolute log-space) respectively; Row 3 show the SDRs (w.r.t. column i reference datasets) of various sound-fields specified on different datasets: reference +, the non-transformed *, and transformed x.

4.5 Conclusions

We have presented a joint spatial-frequency GP fusion method for modeling common-subject sound-fields using HRTFs and linear transformations of HRTFs. Window and equalization transforms are specified an automatically learned for horizontal and median-plane "Club Fritz" HRTF measurements, which characterize inter-dataset measurement process variances. This is verified in experiments where the sound-fields specified on the transformed datasets are much closer to reference sound-fields than non-transformed ones. Future work will consider non-linear transformations between HRTFs over the full sound-field.

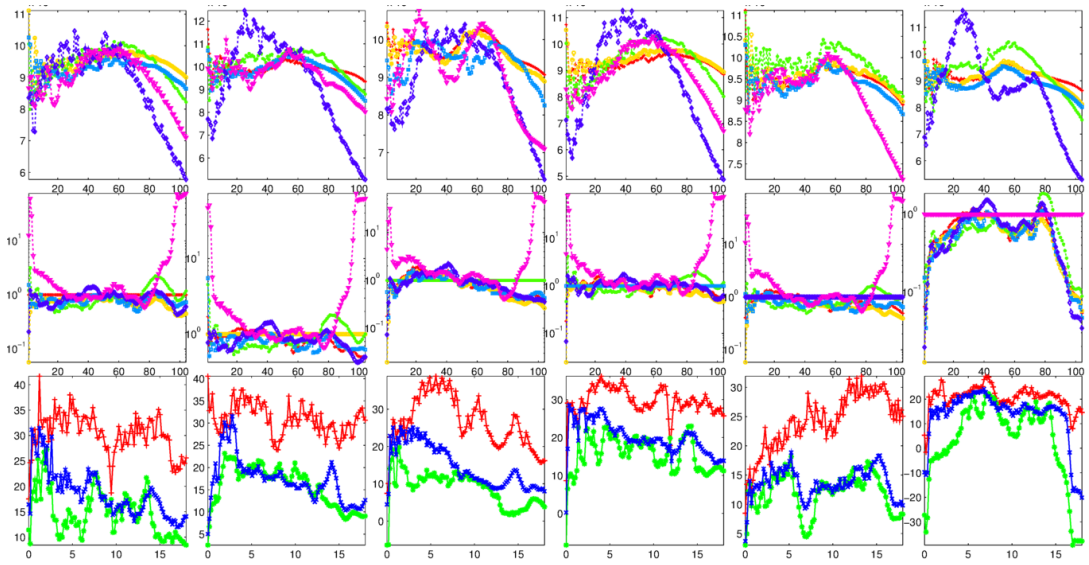


Figure 4.6: Rows 1 and 2 show the median-plane-trained window-filter coefficients (after min-phase reconstruction into time-domain) and the equalization-transform coefficients (absolute log-space) respectively; Row 3 show the SDRs (w.r.t. column i reference datasets) of various sound-fields specified on different datasets: reference +, the non-transformed *, and transformed x.

4.6 Acknowledgment

We thank Dr. B. F. G. Katz and Dr. D. R. Begault for organizing the round-robin activity and hosting the “Club Fritz” datasets.

Chapter 5: Efficient Multicore Non-negative Least Squares

5.1 Introduction

A central problem in data-modelling is the optimization of underlying parameters specifying a linear model used to describe observed data. The underlying parameters of the model form a set n variables in a $n \times 1$ vector $x = \{x_1, \cdots, x_n\}^T$. The observed data is composed of m observations in a $m \times 1$ vector $b = \{b_1, \cdots, b_m\}^T$. Suppose that the observed data are linear functions of the underlying parameters in the model, then the function's values at data points may be expressed as a $m \times n$ matrix A where $Ax = b$ describes a linear mapping from the parameters in x to the observations in b.

In the general case where $m \geq n$, the dense overdetermined system of linear equations may be solved via a least squares approach. The usual way to solve the least squares problem is with the QR decomposition of the matrix A where $A = QR$, with Q an orthogonal $m \times n$ matrix, and R an upper-triangular $n \times n$ matrix. Modern implementations for general matrices use successive applications of the Householder transform to form QR, though variants based on Givens rotation or Gram-Schmidt orthogonalization are also viable. Such algorithms carry an associated $O(mn^2)$ time-complexity. The resulting matrix equation may be rearranged to $Rx = Q^T b$ and x solved via back-substitution.

Sometimes, the underlying parameters are constrained to be non-negative in order

to reflect real-world prior information. When the data is corrupted by noise, the estimated parameters may not satisfy these constraints, producing answers which are not usable. In these cases, it is necessary to explicitly enforce non-negativity, leading to the non-negative least squares (NNLS) problem considered in this paper.

The seminal work of Lawson and Hanson [59] provide the first widely used method for solving this non-negative least squares problem. This algorithm, later referred to as the *active-set method*, partitions the set of parameters or variables into the active and passive-sets. The active-set contains the variables with values forcibly set to zero and which violate the constraints in the problem. The passive-set contains the variables that do not violate the constraint. By iteratively updating a feasibility vector with components from the passive-set, each iteration is reduced to an unconstrained linear least squares sub-problem that is solvable via QR.

For many signal processing applications, NNLS problems in a few hundred to a thousand variables arise. In time-delay estimation for example, multiple systems are continuously stored or streamed for processing. A parallel method for solving multiple NNLS problems would enable on-line applications, in which the estimation can be performed as data is acquired. Motivated by such an application, we develop an efficient algorithm and its implementations on both multi-core CPUs and modern GPUs.

Section 5.1.2 summarizes alternative solutions to the NNLS problem. Section 5.2 establishes notation and formally describes the active-set algorithm. Section 5.3 presents a new method for updating the QR decompositions for the active-set algorithm. Sections 5.4-5.5 describe parallelism on multi-core CPUs and GPU like architectures. Section 5.6 provides a motivating application from remote estimation and section 5.7 compares the

GPU and CPU results from experiments.

5.1.1 Non-negative Least Squares

We formally state the NNLS problem: Given a $m \times n$ matrix $A \in \mathbb{R}^{m \times n}$, find a non-negative $n \times 1$ vector $x \in \mathbb{R}^n$ that minimizes the functional $f(x) = \frac{1}{2}\|Ax - b\|^2$ i.e.

$$\min_x f(x) = \tfrac{1}{2}\|Ax - b\|^2, \qquad x_i \geq 0. \tag{5.1}$$

The Karush-Kuhn-Tucker (KKT) conditions necessary for an optimal constrained solution to an objective function $f(x)$ can be stated as follows [106]: Suppose $\hat{x} \in \mathbb{R}^n$ is a local minimum subject to inequality constraints $g_j(x) \leq 0$ and equality constraints $h_k(x) = 0$, then there exists vectors μ, λ such that

$$\bigtriangledown f(\hat{x}) + \lambda^T \bigtriangledown h(\hat{x}) + \mu^T \bigtriangledown g(\hat{x}) = 0, \quad \mu \geq 0, \quad \mu^T g(\hat{x}) = 0. \tag{5.2}$$

To apply the KKT conditions to the minimization function in Eq. 5.1, the gradient $\bigtriangledown f(x) = A^T(Ax - b)$, $g_j(x) = -x_j$, and $h_k(x) = 0$ leads to the necessary conditions

$$\mu = \bigtriangledown f(\hat{x}), \quad \bigtriangledown f(\hat{x})^T \hat{x} = 0, \quad \bigtriangledown f(\hat{x}) \geq 0, \quad \hat{x} \geq 0, \tag{5.3}$$

that must be satisfied at the optimal solution.

5.1.2 Survey of NNLS Algorithms

A comprehensive review of the methods for solving the NNLS problem can be found in [107]. The first widely used algorithm, proposed by Lawson and Hanson in [59], is the active-set method that we implement on the GPU. Although many newer methods have since surpassed the active-set method for large and sparse matrix systems from our survey, the active-set method remains competitive for *small* to moderate sized systems with unstructured and dense matrices.

In [108], improvements to the original active-set method are developed for the Fast NNLS (FNNLS) variant. By reformulating the normal equations that appear in the pseudo-inverse for the least squares sub-problem, the cross-product matrices $A^T A$ and $A^T b$ can be pre-computed. This contribution leads to significant speed-ups in the presence of multiple right-hand-sides. In [109], further redundant computations are avoided by grouping similar right-hand-side observations that would lead to similar pseudo-inverses.

A second class of algorithms is iterative optimization methods. Unlike the active-set approach, these methods are not limited to a single active constraint at each iteration. In [110], a Projective Quasi-Newton NNLS approach uses gradient projections to avoid pre-computing $A^T A$ and non-diagonal gradient scaling to improve convergence and reduce zigzagging. Another approach in [111] produces a sequence of vectors optimized at a single coordinate with all other coordinates fixed. These vectors have an efficiently computable analytical solution that converge to the solution.

Other methods outside the scope of this review include the Principal Block Pivoting method for large sparse NNLS in [112], and the Interior Point Newton-like method in

[113], [114] for moderate and large problems.

5.2 Active-set Method

Given a set of m linear equations in n unknowns which are constrained to be non-negative, let the active-set Z be the subset of variables which violate the non-negativity constraint or are zero and the passive-set P be the variables with positive values. Lawson and Hanson observe that only a small subset of variables remains in the candidate active-set Z at the solution. If the true active-set Z is known, then the NNLS problem is solved by an unconstrained least squares problem using the variables from the passive-set.

Algorithm 9 Active-set method for non-negative least squares [59]

Require: $A \in \mathbb{R}^{m \times n}$, $x = 0 \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, set $Z = \{1, 2, \ldots, n\}$, $P = \emptyset$
Ensure: Solution $\hat{x} \geq 0$ s.t. $\hat{x} = arg\ min \frac{1}{2}\|Ax - b\|^2$

while true **do**
  Compute negative gradient $w = A^T(b - Ax)$
  if $Z \neq \emptyset$ and $max_{i \in Z}(w_i) > 0$ **then**
    Let $j = arg\ max_{i \in Z}(w_i)$
    Move j from set Z to P
    while true **do**
      Let matrix $A^P \in \mathbb{R}^{m \times *}$ s.t. $A^P = \{\text{columns } A_i \text{ s.t. } i \in P\}$
      Compute least squares solution y for $A^P y = b$
      if $min(y_i) \leq 0$ **then**
        Let $\alpha = -min_{i \in P}(\frac{x_i}{x_i - y_j})$ s.t. (column $j \in A^P$) = (column $i \in A$)
        Update feasibility vector $x = x + \alpha(y - x)$
        Move from P to Z, all $i \in P$ s.t. $x_i = 0$
      else
        Update $x = y$
        break
      end if
    end while
  else
    return x
  end if
end while

In Algorithm 9, the candidate active-set Z is updated by first moving the largest positive component variable in the negative gradient w to the passive-set (line 5). This selects the component with the most negative gradient that reduces the residual 2-norm. The variables in the passive-set form a candidate linear least squares system $A^P y = b$ where matrix A^P contain the column vectors in matrix A that correspond to indices in the passive-set (lines 7, 8). At each iteration, the feasibility vector x moves towards the solution vector y while preserving non-negativity (line 11). Convergence to the optimal solution is proven in [59].

The termination condition (line 3) checks if the gradient is strictly positive or if the residual can no longer be minimized. At termination, the following relations satisfy the optimality conditions in Eq. 5.3:

1. $w_i \leq 0 \quad i \in Z$ termination condition (line 3).

2. $w_i = 0 \quad i \in P$ solving least squares sub-problem (line 8).

3. $x_i = 0 \quad i \in Z$ updating sets (line 12).

4. $x_i > 0 \quad i \in P$ updating x (lines 10-11).

The variables in the passive-set form the corresponding columns of the matrix A^P in the unconstrained least squares sub-problem $A^P y = b$. As discussed previously, the cost of solving the unconstrained least squares sub-problem is $O(mn^2)$ via QR. If there are k iterations, then the cost of k independent decompositions is $O(kmn^2)$. However, the decompositions at each iteration share a similar structure in matrix A^P, and this can be taken advantage of. We observe the following properties of matrix A^P as the iterations

proceed:

1. The active and passive-sets generally exchange a single variable per iteration; one column is added or removed from matrix A^P.

2. Most exchanges move variables from the active-set into the passive-set; early iterations add variables to an empty passive-set to build the feasible solution, while later iterations add and remove variables to refine the solution.

Hence, we develop a general method for QR column updating and downdating that takes advantage of the pattern of movement between variables in the active and passive-sets. To achieve real-time and on-line processing, the method must be parallelizable on GPUs or other multi-core architectures. We note that the improvements made to the active-set NNLS proposed in [108], [109] do not apply to our problem, and moreover do not account for possible efficiencies suggested by the observations above.

5.3 Proposed Algorithm

The first property of matrix A^P suggests that a full $A^P = QR$ decomposition is unnecessary. Instead, we consider an efficient QR column updating and downdating method.

1. QR Updating: A new variable added to set P expands matrix A^P by a single column. Update previous matrices Q, R with this column insertion.

2. QR Downdating: The removal of a variable from set P shrinks matrix A^P by a single column. Downdate previous matrices Q, R with this column deletion.

The second property of matrix A^P suggests that we can optimize the cost for QR updating in terms of floating point operations (flops) and column or row memory accesses. We observe that many QR updating methods minimize computations when inserting columns at the right-most index. Our method takes advantage of this by maintaining a separate ordering for the columns of matrix A^P by the relative times of insertions and deletions across iterations. That is, a column insertion always appends to the end of a reordered matrix $\hat{A^P}$. We describe the effects of the reordering strategy for various updating methods in sections 5.3.1-5.3.3. We also show that the modified Gram-Schmidt and Givens rotation methods are the most cost efficient with respect to the reordering strategy for overdetermined and square systems.

5.3.1 QR Updating by Modified Gram-Schmidt

The reordering strategy allows a new column a_i from the matrix $A^P = [a_1, \cdots, a_i, \cdots, a_n]$ to be treated as the right-most column in the decomposition. We define list $\hat{P}$ as an ordered list of column indices from set P such that the associated column $\hat{p}_{i-1}$ is added in a prior iteration to column $\hat{p}_i$. The reordered decomposition $\hat{A^P} = \hat{Q}\hat{R}$ is

$$\hat{A^P} = [a_{\hat{p}_1}, \cdots, a_{\hat{p}_{i-1}}, a_{\hat{p}_i}], \quad \hat{Q} = [q_{\hat{p}_1}, \cdots, q_{\hat{p}_{i-1}}, q_{\hat{p}_i}], \quad \hat{R} = [r_{\hat{p}_1}, \cdots, r_{\hat{p}_{i-1}}, r_{\hat{p}_i}], \tag{5.4}$$

where $\hat{Q}$ is a $m \times i$ matrix and $\hat{R}$ is an $i \times i$ matrix. To compute column $q_{\hat{p}_i}$, we orthogonalize the inserted column a_i with all the previous columns in matrix $\hat{Q}$ via vector projections. To compute column $r_{\hat{p}_i}$, we take the inner products between column a_i and

columns in $\hat{Q}$, or the equivalent matrix-vector product $\hat{Q}^T a_i$. Both quantities are found using the Modified Gram-Schmidt (MGS) procedure in Algorithm 10.

Algorithm 10 Reordered MGS QR Column Updating

Require: Reordered list $\hat{P}$ contains the elements in set P, index i the variable added to set P, column a_i the new column in A^P, columns $q_j \in Q$

Ensure: $\hat{A^P} = \hat{Q}\hat{R}$, update vector $\hat{Q}^T b$, list $\hat{P}$

1: Let vector $u = a_i$
2: **for all** column index $k \in$ list $\hat{P}$ **do**
3: $\quad u = u - \langle q_k, u \rangle q_k$
4: $\quad \hat{R}_{ki} = \langle a_i, q_k \rangle$
5: **end for**
6: $q_i = \frac{u}{\|u\|}$
7: $\hat{R}_{ii} = \|u\|$
8: $\hat{Q}^T b_i = \langle q_i, b \rangle$
9: Add i to list $\hat{P}$

With the reordering strategy in Algorithm 10, a new column a_i is always inserted in the right-most position of matrix $\hat{A^P}$. The number of columns read from memory in matrix $\hat{Q}$ is the size of set P, denoted as $\ell \leq n$ and is used to form column q_i. The number of column memory writes per step is two, as column q_i appends to matrix $\hat{Q}$ and the projection step writes a single column to matrix $\hat{R}$. Updating matrices $\hat{Q}$ and $\hat{R}$ requires $6m\ell + 3m + 1$ flops. The asymptotic complexity is $O(mn)$.

Without the reordering strategy, column a_i can be inserted into the middle of matrix $\hat{A^P}$. This requires computing column q_i the re-orthogonalization of the $\ell - i$ columns to its right. The memory access costs of computing q_i is i number of columns reads from matrix $\hat{Q}$ and two columns writes to matrices $\hat{Q}$ and $\hat{R}$. The re-orthogonalization costs of column q_j where $j > i$ is equivalent to a new column insertion into matrix A^P. This is because the MGS method does not compute the null-space of the basis vectors in matrix $\hat{Q}$. Orthogonalizing columns q_j and q_{j+1} with respect to column q_i does not preserve the

orthogonality between q_j and q_{j+1}. Thus, each of the $\ell - i + 1$ columns must be reinserted with an additional $\ell(\ell - i + 1)$ column reads and $2(\ell - i + 1)$ column writes. Updating matrices $\hat{R}$ and $\hat{Q}$ requires a total of $(3m\ell + 3mi + 1)(\ell - i + 2)$ flops. The asymptotic complexity is $O(mn^2)$.

5.3.2 Alternative QR Updating by Rotations

Rotation based methods for updating QR are possible. In [115], $\hat{Q}$ and $\hat{R}$ are treated as $m \times m$ matrices where matrix $\hat{Q}$ is initially the identity. When inserting column a_i, the method appends $m \times 1$ column vector $r_{\hat{p}_i} = \hat{Q}^T a_i$ to matrix $\hat{R}$. A series of rotation transformations introduces zeros to rows $\{i + 1, i + 2, \cdots, m\}$ of column $r_{\hat{p}_i}$ to preserve the upper-triangular property. The rotation transformations then update the columns to the right of index i in matrices $\hat{R}$ and $\hat{Q}$. A similar step follows updating the right-hand side $\hat{Q}^T b_i$.

Without the reordering strategy, the costs of this rotation method depend on index i. Column $r_{\hat{p}_i}$ requires $m - i$ rotation transformations. Each transformation requires two row memory reads and writes to matrix $\hat{R}$ and two column memory reads and writes to matrix $\hat{Q}$ for a total of $2(m - i)$. This is disadvantageous as the number of column and row accesses is bound by m and multiple columns and rows of matrices $\hat{R}$ and $\hat{Q}$ are modified. Updating matrix $\hat{Q}$ and $\hat{R}$ requires $6m(m - i)$ and $2m^2 + 8(m - i) + 6(\ell - i + 1)(\ell/2 - i/2 - 1)$ flops respectively. The asymptotic complexity is $O(m^2 + n^2)$.

With the reordering strategy, index $i = \ell + 1$ and so many of the costs are reduced. There are no columns to the right of index i in matrix $\hat{R}$ so updating is limited to single

column memory write of column $r_{\hat{p}_i}$. Updating matrix $\hat{Q}$ now requires $m - \ell - 1$ column reads and writes each while applying the transformations. Updating matrices $\hat{Q}$ and $\hat{R}$ requires $6m(m - \ell - 1)$ and $2m^2 + 8(m - \ell - 1)$ flops respectively. The asymptotic complexity is $O(m^2)$.

5.3.3 Alternative QR Updating by Semi-normal Equations

The corrected semi-normal equations (CSNE) can be used to update a $\ell \times \ell$ matrix $\hat{R}$ without the construction of matrix $\hat{Q}$. The stability analysis of this method is provided by [116]. With the reordering strategy, the problem treats $\hat{R}^T\hat{R}x = \hat{A^P}b$ where column $r_{\hat{p}_i}$ is computed by

$$\hat{R}^T\hat{R}z = \hat{A^P}a_i, \quad s = a_i - \hat{A^P}z, \quad \hat{R}^T\hat{R}\delta z = s, \quad z = z + \delta z, \quad r_{\hat{p}_i} = \begin{bmatrix} \hat{R}z \\ \|\hat{A^P}z - a_i\| \end{bmatrix}. \tag{5.5}$$

Although the method does not compute and store matrix $\hat{Q}$, it requires both row and column access to matrix $\hat{R}$ and more operations to produce column $r_{\hat{p}_i}$. Computing and correcting for vector z entails four back-substitutions using matrices $\hat{R}^T$, $\hat{R}$, and $\hat{A^P}$. All four back-substitutions requires ℓ row or column memory reads from matrix $\hat{R}$ each. Two of the back-substitutions require m row memory reads from matrix $\hat{A^P}$. The total number of column and row memory reads in the method is $3m + 2\ell$ and one column memory write to update matrix $\hat{R}$. The entire procedure requires $6m^2 + 4m + 1 + 3\ell^2 + 3\ell$ flops. The asymptotic complexity is $O(m^2 + n^2)$.

Without the reordering strategy, the same CSNE method computes column $r_{\hat{p}_i}$ and a series of rotations introduces zeros below index i. The rotation transformations are then applied to the columns to the right of index i in matrix $\hat{R}$. This requires an additional $\ell - i$ row memory reads and writes to matrix $\hat{R}$ each and $6(\ell - i + 1)(\ell/2 - i/2 - 1) + 3(\ell - i)$ flops. The asymptotic complexity is $O(n^2)$. The costs for the updates are summarized in Table 5.1.

5.3.4 QR Downdating by Rotations

The reordering strategy is less applicable to the downdating scheme as deleted columns may not be in the right-most index. Suppose that column a_i is removed from matrix $A^P = [a_1, \cdots, a_{i-1}, a_{i+1}, \cdots, a_n]$. Let $\hat{p}_j$ be the corresponding column index in the ordered list. We consider the reformulation of Eq. 5.4 without column $\hat{p}_j$ as

$$\begin{aligned} \tilde{A^P} &= [a_{\hat{p}_1}, \cdots, a_{\hat{p}_{j-1}}, a_{\hat{p}_{j+1}}, \cdots, a_{\hat{p}_i}], \quad \tilde{Q} = [q_{\hat{p}_1}, \cdots, q_{\hat{p}_{j-1}}, q_{\hat{p}_j}, q_{\hat{p}_{j+1}}, \cdots, q_{\hat{p}_i}], \\ \tilde{R} &= [r_{\hat{p}_1}, \cdots, r_{\hat{p}_{j-1}}, r_{\hat{p}_{j+1}}, \cdots, r_{\hat{p}_i}], \end{aligned} \tag{5.6}$$

where $\tilde{A^P} = \tilde{Q}\tilde{R}$, matrices $\tilde{A^P}$ and $\tilde{R}$ are missing column $\hat{p}_j$, and matrix $\tilde{Q} = \hat{Q}$ is unchanged. Column $q_{\hat{p}_j}$ still exists in matrix $\tilde{Q}$ and matrix $\tilde{R}$ is no longer upper-triangular as the columns to right of index $\hat{p}_j$ have shifted left.

Observe that right sub-matrix shifted in matrix $\tilde{R}$ has a Hessenberg form. In [117], a series of Givens rotations introduces zeros along the sub-diagonal. However, this does not directly address the removal of column $\hat{p}_j$ in matrix $\tilde{Q}$. Instead, we apply a series of Given rotations to introduce zeros along the j^{th} row of matrix $\tilde{R}$. The rotations are

applied to the right of column $\hat{p}_j$ in the transformation

$$\tilde{A^P} = \tilde{Q}(\prod_{k=j}^{i} G_k^T)(\prod_{k=i}^{j+1} G_k)\tilde{R}, \quad \left(\prod_{k=i}^{j+1} G_k\right)\tilde{R} = \begin{bmatrix} \ddots & & & & & & \\ & * & * & * & * & & \\ & 0 & * & * & * & & \\ \cdots & 0 & 0 & 0 & 0 & \cdots & \\ & 0 & 0 & * & * & & \\ & 0 & 0 & 0 & * & & \\ & & & & & \ddots & \end{bmatrix}, \tag{5.7}$$

which preserves $\tilde{A^P} = \tilde{Q}\tilde{R}$ while introducing zeros along the j^{th} row of matrix $\tilde{R}$ and modifying matrix $\tilde{Q}$. This enables both row j in the updated matrix $\tilde{R}$ and column j in updated matrix $\tilde{Q}$ to be removed without violating matrix $\tilde{A^P}$. Vector $\tilde{Q}^T b$ is updated via a similar transformation.

Algorithm 11 Reordered QR Column Downdating with Givens Rotations

Require: Reordered list $\hat{P}$ contains the elements in set P, index i is the variable removed from set P

Ensure: $\tilde{A^P} = \tilde{Q}\tilde{R}$, update vector $\tilde{Q}^T b$, list $\hat{P}$

1: **for all** column indices k following ($\succ$) index $i \in$ list $\hat{P}$ **do**

2: Let $r = \sqrt{\tilde{R}_{kk}^2 + \tilde{R}_{ik}^2}, \quad c = \frac{\tilde{R}_{ik}}{r}, \quad s = \frac{\tilde{R}_{kk}}{r}$

3: $$\begin{bmatrix} \tilde{R}_{k,j \succeq k \in \hat{P}} \\ \tilde{R}_{i,j \succeq k \in \hat{P}} \end{bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} \tilde{R}_{k,j \succeq k \in \hat{P}} \\ \tilde{R}_{i,j \succeq k \in \hat{P}} \end{bmatrix}$$

4: $$\begin{bmatrix} \tilde{Q}_{:,k} & \tilde{Q}_{:,i} \end{bmatrix} = \begin{bmatrix} \tilde{Q}_{:,k} & \tilde{Q}_{:,i} \end{bmatrix} \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$$

5:

6: Let coefficient $b\_h = \tilde{Q}^T b_k, \quad b\_l = \tilde{Q}^T b_i$

7: Set $\tilde{Q}^T b_k = c * b\_h + s * b\_l, \quad \tilde{Q}^T b_i = -s * b\_h + c * b\_l$

8: **end for**

9: Remove index i from list $\hat{P}$

We refer to [117] for precautions when computing the rotation coefficients c, r in Algorithm 11. When updating matrices $\tilde{R}$ and $\tilde{Q}$, a row or column is fixed so the transformation requires $2(\ell - i + 1)$ row and column memory reads and writes each. Updating matrices $\tilde{Q}$ and $\tilde{R}$ requires $6m(m-i)$ and $6(\ell-i)+6(\ell-i+1)(\ell/2-i/2-1)$ flops respectively. The asymptotic complexity is $O(m^2+n^2)$. The costs for the downdates are summarized in Table 5.1.

Algorithm	Col/row accesses	Up/down Q flops	Up/down R flops
MGS/up/reorder	$\ell + 2$	$6m\ell + 3m + 1$	included in Q
MGS/up/unorder	$\ell(\ell - i + 2) + 2(\ell - i + 2)$	$(3m\ell + 3mi + 1)(\ell - i + 2)$	included in Q
Rot/up/reorder	$2(m - \ell - 1)$	$6m(m - \ell - 1)$	$2m^2 + 8(m - \ell - 1)$
Rot/up/unorder	$4(m - \ell - 1)$	$6m(m - i)$	$2m^2 + 8(m - i) + 6(\ell - i + 1)(\ell/2 - i/2 - 1)$
CSNE/up/reorder	$3m + 2\ell + 1$	NA	$6m^2 + 4m + 1 + 3\ell^2 + 3\ell$
CSNE/up/unorder	$3m+4\ell-2i+1$	NA	$6m^2+4m+1+3\ell^2+6(\ell-i+1)(\ell/2-i/2-1)+6\ell-3i$
Rot/down/NA	$4(\ell - i + 1)$	$6m(m - i)$	$6(\ell-i)+6(\ell-i+1)(\ell/2-i/2-1)$

Table 5.1: Costs for QR updating/downdating methods with respect to the reordering strategy. The rotation and CSNE methods have flops of order m^2. For overdetermined and square systems where $\ell \leq n \leq m$, this quantity is minimized for the modified Gram-Schmidt method.

5.4 Multi-core CPU Architectures

The multi-core trend began as a response to the slowdown of Moore's Law while manufactures approached the limitations in single-core clock speeds. With additional cores added on chip, individual CPU threads can be assigned and processed by their own units in hardware. Thus, a single problem is decomposed and solved by several threads without over-utilizing a single core. This gave multi-threading an edge over traditional single-core processors as data and instruction level caches could be dedicated to a smaller sub-set of

operations.

Such Multiple-Instruction-Multiple-Data (MIMD) architectures support task-level parallelism where each core can asynchronously execute separate threads on separate data regions. The individual cores are often super-scalar and thus capable of processing out-of-order instructions in their pipeline. This allows multi-core architectures to simulate data-level parallelism from Single-Instruction-Multiple-Data (SIMD) like architectures such as the GPU with added proficiency. Furthermore, multi-core architectures have access to a common pool of main memory off-die and capable of multi-level caching per core and per processor on-die. For both data and task-level parallelism, this allows memory to be decomposed and cached on a per-core basis for efficient reuse.

Several application programming interfaces (APIs) and libraries take advantage of these shared memory multiprocessing environments for high performance computing. Open Message Passing (OpenMP) is an API based on fork-join operations where the program enters into a designated parallel region [60]. Each thread exhibits both task and data-level parallelism as it independently executes code within a same parallel region. The Intel Math Kernel Library (MKL) is a set of optimized math routines with calls to Basic Linear Algebra Sub-programs (BLAS) and Linear Algebra PACKage (LAPACK) libraries [118]. Many of its fundamental matrix and vector routines are blocked and solved across multiple threads.

5.4.1 CPU Implementation

To exploit the advantages of multi-threading, we adopt both the OpenMP API and the Intel MKL in the CPU implementation. One way to map each linear system to a thread is to declare the entire NNLS algorithm within a parallel OpenMP region. That is, a specified fraction of threads execute NNLS on a mutually exclusive set of linear system of equations. The remaining threads are dedicated to the MKL library in order to accelerate common matrix-vector and vector-vector operations used to solve the unconstrained least squares sub-problem.

5.5 GPU Architectures

Recent advances in general purpose graphics processing units (GPUs) have given rise to highly programmable architectures designed with parallel applications in mind. Moreover, GPUs are considered to be typical of future generations of highly parallel, multi-threaded, multi-core processors with tremendous computational horsepower. They are well-suited for algorithms that map to a Single-Instruction-Multiple-Thread (SIMT) architecture. Hence, GPUs achieve a high arithmetic intensity (ratio of arithmetic operation to memory operations) when performing the same operations across multiple threads on a multi-processor.

GPUs are often designed as a set of multiprocessors, each containing a smaller set of scalar-processors (SP) with a Single-Instruction-Multiple-Data (SIMD) architecture. Hardware multi-threading under a SIMT architecture maps multiple threads to a single SP. A single SP handles the instruction address and register states of multiple threads so

that they may execute independently. The multiprocessor's SIMT unit schedules batches of threads to execute a common instruction. If threads of the same batch diverge via a data-dependent conditional branch, then all the threads along the separate branches are serialized until they converge back to the same execution path.

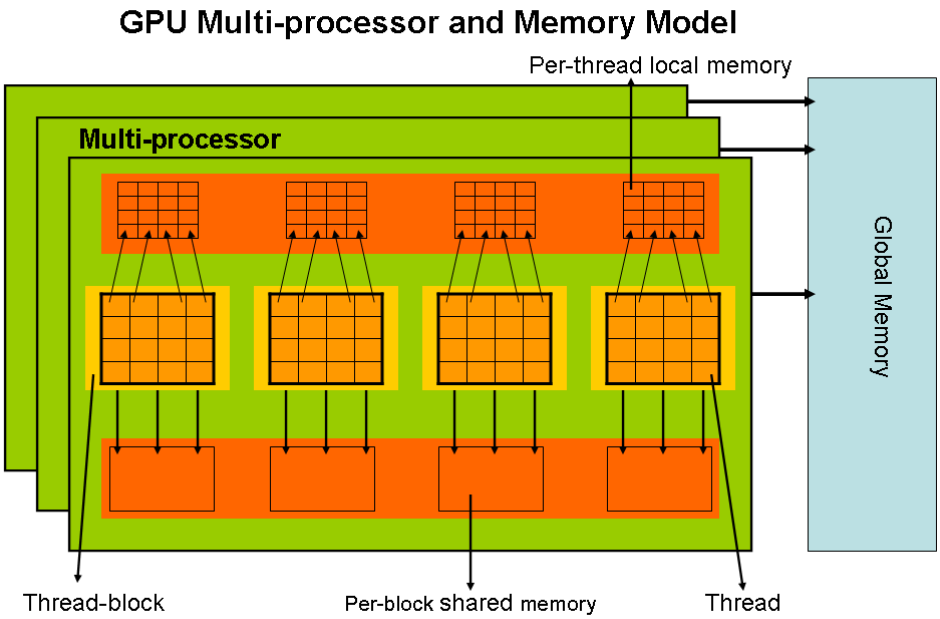


Figure 5.1: GPU multiprocessor utilizes hierarchical memory model spanning fast on-chip and shared memory accessible by the local multiprocessor to slow off-ship global memory accessible by all multiprocessors.

GPUs have a hierarchical memory model with significantly different access times to each level. At the top, all multiprocessors may access a global memory pool on the device. This is the common space where input data is generally copied and stored from main memory by the host. It is also the slowest memory to access as a single query from a multi-processor has a 400 to 600 clock cycles latency on a cache-miss. See [61] for a discussion on coalesced global memory accesses which reads or writes to a continuous chunk of memory at a cost of one query and implicit caching on the Fermi architecture.

On the same level, texture memory is also located on the device but can only be written to from hosts. However, it is faster than global memory when access patterns are spatially local. On the next level, SPs on the same multi-processor have access to a fast shared memory space. This enables explicit inter-thread communication and temporary storage for frequently accessed data. Constant memory, located on each multi-processor, are cached and optimized for broadcasting to multiple threads. On the lowest level, a SP has its own private set of registers distributed amongst its assigned threads. The latency for accessing both shared and per-processor registers normally adds zero extra clock cycles to the instruction time. See Figure 5.1 for a visualization.

Programming models such as NVIDIA's Compute Unified Device Architecture (CUDA) [61] and OpenCL [119] organize threads into thread-blocks, which in turn are arranged in a 2D grid. A thread-block refers to a 1D or 2D patch of threads that are executed on a single multiprocessor. These threads efficiently synchronize their instructions and pass data via shared memory. Instructions are generally executed in parallel until a conditional branch or an explicit synchronization barrier is declared. The synchronization barrier ensures that the thread-block waits for all its threads to complete its last instruction. Thus, two levels of data parallelism are achieved. The threads belonging to the same thread-block execute in lock-step as they process a set of data. Individual thread-blocks execute asynchronously but generally with the same set of instructions on a different set of data.

While efficient algorithms on sequential processors must reduce the number of computations and cache-misses, parallel algorithms on GPUs are more concerned with minimizing data dependencies and optimizing accesses to the memory hierarchy. Data

dependency increases the number of barrier synchronizations amongst threads and is often subject to the choice of the algorithm. Memory access patterns present a difficult bottleneck on multiple levels. While latency is the first concern for smaller problems, we run into a larger issue with memory availability as the problem size grows. That is, the shared memory and register availability are hard limits that bound the size and efficiency of thread-blocks. A register memory bound per SP limits the number of threads assigned to each SP and so decreases the maximum number of threads and thread-blocks running per multi-processor. A shared memory bound per multi-processor limits the number of thread-blocks assigned to a multi-processor and so decreases the total number of threads processed per multi-processor.

5.5.1 GPU Implementation

One way to map each linear system onto a GPU is to consider every thread-block as an independent vector processor. Each thread-block of size $m \times 1$ maps to the elements in a column vector and asynchronously solves for a mutually exclusive set of linear systems. The number of thread-blocks that fit onto a single multi-processor depends on the column size m or the number of equations in the linear system. This poses a restriction on the size of linear systems that our GPU implementation can solve as the maximum size m is constrained to a fraction of the amount of shared memory available per multi-processor. Fortunately, this is not an issue for applications where m is small (500-1000) and the number of linear systems to be solved is large. However for arbitrarily sized linear systems of equations, our GPU implementation is not generalizable. We note that this is not

an algorithmic constraint but rather a design choice for our application. Our multi-core CPU implementation of the same algorithm can solve for arbitrarily sized linear systems. We discussion the details of the GPU implementation in sections 5.5.2-5.5.3.

5.5.2 Parallelizing QR Methods

Full QR decompositions on the GPU via blocked MGS, Givens rotations, and Householder reflections are implemented in [120], [121]. While [121] cites that the blocked MGS and Givens rotation methods are ill-suited for large systems on GPUs as they suffer from instability and synchronization overhead, we are interested in only the QR updating and downdating schemes for a large number of *small* systems. We show that it is possible for m threaded multi-processors to efficiently perform the MGS updating and Givens rotation downdating steps.

For the MGS update step, most of the operations are formulated as vector inner products, scalar-vector products, and vector-vector summations. These operations lead to an one-to-one mapping between the $m \times 1$ column vector coordinates and the m threaded thread-block. Such operations are computable via parallel reduction techniques from [122]. In algorithm 10, we parallelize all four inner products (lines 3, 4, 6, 8) in $\log m$ parallel time each. The inner loop iterates for ℓ or at most n times. Thus, we obtain an order reduction in parallel time-complexity to $O(n \log m)$.

For the Givens rotation downdate step, we obtain an one-to-many mapping between the $n \times 1$ row vector elements and the m threads in a thread-block for matrix R. We obtain an one-to-one mapping for the $m \times 1$ column vector elements in matrix Q. Computing

vector $Q^T b$ follows a similar relation. For obtaining the rotation coefficients c, s, a single thread computes and broadcasts to the rest of the thread-block. In Algorithm 11, the inner loop (lines 3-4) updates both matrices $\tilde{R}$ and $\tilde{Q}$ in parallel $O(1)$ time. Writing row and column data and updating vector $\tilde{Q}^T b$ (line 7) are thread-independent and computable in $O(1)$ parallel time. Thus, we obtain an order reduction in parallel time-complexity to $O(n)$.

Parallel reductions are often performed on the GPU in place of common vector-vector operations using prefix sum discussed in [123]. Algorithm 12 sums 512 elements in 9 parallel flops, 5 thread-synchronizations, and 18 parallel shared memory accesses. Each of the 512 threads reserves a memory slot in shared memory. The unique thread ID or tID denotes the corresponding data index in the shared memory array. At each step, half the threads from the previous step sum up the data entries stored in the other half of shared memory. The process continues until index 0 in the shared memory array contains the total summation.

5.5.3 Memory Usage

To take advantage of different access times on the GPU memory hierarchy, the input and intermediate data can be stored and accessed on different levels for efficient reuse. Local intermediate vectors can either be stored in shared memory or alternatively in dedicated registers spanning all threads in a thread-block. List $\hat{P}$ is stored in shared memory as multiple threads require synchronization to update and downdate the same column. The right-hand-side vector $\hat{Q}^T b$ is stored in registers since no thread accesses elements outside

Algorithm 12 CUDA parallel floating-point summation routine [123]

```
__device__ float reduce512( float smem512[], unsigned short tID){
        __syncthreads();
        if(tID < 256)   smem512[tID] += smem512[tID + 256];
        __syncthreads();
        if(tID < 128)   smem512[tID] += smem512[tID + 128];
        __syncthreads();
        if(tID < 64)    smem512[tID] += smem512[tID + 64];
        __syncthreads();
        if(tID < 32){
                smem512[tID] += smem512[tID + 32];
                smem512[tID] += smem512[tID + 16];
                smem512[tID] += smem512[tID + 8];
                smem512[tID] += smem512[tID + 4];
                smem512[tID] += smem512[tID + 2];
                smem512[tID] += smem512[tID + 1];
        }
        __syncthreads();
        return smem512[0];
}
```

its one-to-one mapping in the update and downdate steps.

Global memory accesses on the GPU are unavoidable for updating large matrices $\hat{Q}$ and $\hat{R}$. We store matrix $\hat{Q}^T$ so that column vector accesses are coalesced in row-oriented programming models and matrix $\hat{R}$ as the Given rotations update the rows. Matrices $\hat{Q}$ and $\hat{R}$ are stored in-place unlike the compact format in Eqs. 5.4, (5.6). We allocate $m \times n$ blocks of global memory and use the reordered list $\hat{P}$ to associate column and row indices for the update and downdate steps. This is to avoid any physical shifts of column vectors in global memory. Rather, we parallel shift the list $\hat{P}$ when a variable is removed from the passive-set.

The MGS update step reads ℓ number of columns in matrix $\hat{Q}$ from global memory into registers. Computing inner products and vector norms during the projections requires

an intermediate shared memory vector for the parallel reduction function. The new column for matrix $\hat{R}$ is locally stored in registers before updated to global memory. A single element for vector $\hat{Q}^T b$ is updated and written to shared memory. The total number of parallel shared memory accesses is $39\ell + 2$. The total number of parallel global memory accesses is $\ell + 2$.

The Givens rotations downdate step accesses two columns of matrix $\tilde{Q}$ and two rows of matrix $\tilde{R}$ for each of the $\ell - i$ transformation. Since row i of matrix $\tilde{R}$ and column i of matrix $\tilde{Q}$ are fixed across transformations, they are stored and updated in shared memory. The other row and column are directly updated in global memory. Updating vector $\hat{Q}^T b$ requires two shared memory reads and writes. The total number of parallel shared memory accesses is $2(\ell - i) + 2$. The total number of parallel global memory accesses is $2(\ell - i + 1)$.

5.6 Applications for Deconvolution and Time-Delay Estimation

In remote sensing, a discrete-time deconvolution recovers a signal x that has been convolved with a transfer function s. The known signal s is often convolved with an unknown signal x that satisfies properties such as non-negativity. The deconvolution problem can modeled as

$$y(t) = s(t)x(t) = \int_{-\infty}^{\infty} s(\tau)x(t-\tau)\,d\tau = \int_{-\infty}^{\infty} x(\tau)s(t-\tau)\,d\tau \approx \sum_{\tau=1}^{n} x(\tau)s(t-\tau)\,d\tau, \tag{5.8}$$

where t is the sample's time, $y(t)$ the observed signal, and n the number of samples over time. To solve for the unknown signal x, we rewrite Eq. 5.8 as the following square linear system of equations $Ax = b$, where A is a Toeplitz matrix:

$$A = \begin{bmatrix} s(0) & s(-1) & \cdots & s(-(n-1)) \\ s(1) & s(0) & \cdots & s(-(n-2)) \\ \vdots & \vdots & \ddots & \vdots \\ s(n-1) & s(n-2) & \cdots & s(0) \end{bmatrix}, \quad x = \begin{bmatrix} x(1) \\ \vdots \\ x(n) \end{bmatrix}, \quad b = \begin{bmatrix} y(1) \\ \vdots \\ y(n) \end{bmatrix}. \tag{5.9}$$

Efficient algorithms for the deconvolution problem, which either exploit the simple structure of the convolution in Fourier space, or which exploit the Toeplitz structure of the matrix, are available in [124], [125]. However, if signal x is known to be non-negative and the data $y(t)$ is corrupted by noise, then we may treat the deconvolution as a NNLS problem.

A similar problem arises in time-delay estimation between a common audio source signal recorded at different points in space. Knowledge of multiple time-delay estimations can be used to localize sound in a spherical domain. In the case of human sound localization, the inter-aural time difference (ITD) between left-right ear sound measurements s_l, s_r can be formulated as the following overdetermined Toeplitz system of linear

equations:

$$A = \begin{bmatrix} s_r(0) & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ s_r(n-1) & \cdots & \cdots & \vdots \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & s_r(0) \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & s_r(n-1) \end{bmatrix}, \quad x = \begin{bmatrix} x(1) \\ \vdots \\ x(2n) \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ s_l(0) \\ \vdots \\ s_l(n-1) \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{5.10}$$

where matrix $A \in \mathbb{R}^{3n \times 2n}$ is the first $2n$ columns of a Toeplitz matrix with leading column $[s_r(0), \ldots, s_r(n-1), 0 \ldots 0]^T \in \mathbb{R}^{3n \times 1}$ and zero-row vectors; samples of signal s_r are shifted w.r.t. the column index and zero padded. Vector b is the signal s_l nested between zero-column vectors of size n. Thus, solution vector x is the set of non-negative weights for the linear combination of time-delayed signals s_r that best reconstructions signal s_r in the least squares sense.

5.7 Experiments

As a baseline, we note that Matlab's lsqnonneg function implements the same active-set algorithm but with a full QR decomposition for the least squares sub-problem. Matlab

2009b and later versions use Intel's Math Kernel Library (MKL) with multi-threading to resolve the least squares sub-problems. For a better comparison, we first port the lsqnonneg function into native C-code with calls to multi-threaded MKL BLAS and LAPACK functions. The results from this implementation (CPU lsqnonneg) show a 1.5-3x speed-up over the Matlab lsqnonneg function in our experiments. Next, we apply our updating and downdating strategies with column reordering using MKL BLAS functions to the CPU version. The results from this second implementation (CPU NNLS) show a 1-3x speed-up that depends on the number of column updates and downdates. Last, we compare the lsqnonneg variants to alternative NNLS algorithms from literature.

To compare GPU implementation with the multi-threaded CPU variants, we begin timing the point of entry and exit out of the GPU kernel function. Memory transfer and pre-processing times in the case of non-synthetic data are omitted. Both the GPU and CPU variants also obtain identical solutions subject to rounding error within the same number of iterations for all data sets. We find that for a fewer number of linear systems, the CPU implementations outperforms the GPU as only a fraction of the GPU cores are utilized. When the number of linear systems surpasses the number of multi-processors, the GPU scales better on an order of 1-3x than our fastest CPU implementation.

For reference, we use a Dual Quad-Core Intel(R) Xeon(R) X5560 CPU @ 2.80GHz (8 cores) for testing our CPU implementations. The CPU codes compiled under both Intel icc 11.1, gcc 4.5.1, and linked to MKL 10.1.2.024 yield similar results for 8 run-time threads. The codes tested between Matlab 2010b and 2009b also yield comparable results. Mixing the number of threads assigned between OpenMP and MKL did not have a large impact on our system. We use a NVIDIA Tesla C2050 (448 cores across 14

multi-processors) and codes compiled under CUDA 3.2 for the GPU implementation and testing.

5.7.1 Synthetic Deconvolution

For the first set of synthetic data, we generate mean shifted 1-D Gaussians with $\sigma = 4.32$ to store as columns in matrix A of size $m = n = 512$. In this Gaussian fitting problem, each system uses the same matrix A but with non-negative random vectors b. The choice of the σ parameter ensures that the mean shifted Gaussians are not too wide as to allow early convergence and not too narrow as to locally affect only a few variables. Furthermore, matrix A is now considered dense and vectors b no longer reflect real-world values. We expect the average number of iterations or column updates and downdates to exceed that of the real-world data cases.

The total speed-up of GPU NNLS over the CPU variants are more pronounced (3x compared to CPU NNLS, 23x compared to CPU lsqnonneg). The larger ratio of column downdate to update steps suggests that our reordering strategy and fast Givens rotation method in the downdating step outperforms the lsqnonneg variants.

For the second set of synthetic data, we generate both random matrices A of size $m = n = 512$ and non-negative random vectors b. The number of column updates and downdates is less than that of the two previous experiments. Furthermore, the total number of column updates dominates the number of column downdates. The results between GPU and CPU NNLS show that both implementations have similar run-time scaling as the number of systems increases. This suggests that the most of the performance gains in

Number of systems	1	24	48	96	192
Number of updates	165	3907	7792	15541	30966
Number of downdates	92	2109	4196	8362	16599
GPU NNLS	0.4257	0.5094	0.9546	1.7862	3.2672
CPU NNLS	0.0654	1.2238	2.4141	4.8067	9.6250
CPU lsqnonneg	0.4791	8.7611	17.2483	34.5396	69.4889
Matlab lsqnonneg	0.9437	19.5904	38.6469	77.1072	155.7700
Matlab FNNLS [108]	0.4937	11.7176	23.9502	47.4635	91.4500
Matlab interior-points [114]	1.6317	40.7017	83.9788	164.4839	328.9569
Matlab PQN-NNLS [110]	3.1106	128.6616	253.3796	504.3153	989.2051

Table 5.2: Runtime (seconds) comparisons of NNLS and lsqnonneg variants on multiple systems of mean shifted Gaussian matrix A and random vectors b.

prior experiments are from the GPU downdating steps. The results between CPU NNLS and CPU lsqnonneg leads to a similar conclusion as the performance gain (1.7x) is minimum compared to the prior two experiments.

Number of systems	1	24	48	96	192
Number of updates	52	1169	2361	4766	9525
Number of downdates	0	13	28	58	124
GPU NNLS	0.1194	0.1397	0.2526	0.4875	0.8667
CPU NNLS	0.0068	0.1157	0.2245	0.4352	0.8794
CPU lsqnonneg	0.0223	0.4665	0.8959	1.7269	3.5243
Matlab lsqnonneg	0.0330	0.8096	1.5583	3.0097	6.1627
Matlab FNNLS [108]	0.0248	0.5902	1.1602	2.2920	4.6022
Matlab interior-points [114]	0.4480	10.7729	21.1767	41.9441	83.6695
Matlab PQN-NNLS [110]	1.5935	55.8196	110.0892	221.7277	441.7190

Table 5.3: Runtime (seconds) comparisons of NNLS and lsqnonneg variants on multiple systems of random matrices A and random vectors b.

5.7.2 Non-Synthetic Deconvolution

For real-world data, we use terrain laser imaging sets obtained from the NASA Laser Vegetation Imaging Sensor (LVIS)1. Each data set contains multiple 1-D Gaussian-like

signals s and observations of total return energy b of size $m = n = 432$. In this deconvolution problem, the transfer functions s represent the single impulse energy fired over time on ground terrain and the observed signals b produces a waveform that indicate the reflected energy over time. The signals s are generally 15-25 samples wide so the computed matrices A are Toeplitz banded and sparse. NNLS solves for corresponding pairs of matrix A and vector b to obtain the sparse non-negative solutions x that represent the times of arrival for a series of a fired impulses. This estimates the ranges or distances to a surface target.

For comparing the NNLS methods, we record the run-times in relation to the number of column updates and downdates for the least squares sub-problem. The results from CPU NNLS show a 11x speed-up over the GPU implementation when solving for a single system. This is due to the underutilization of cores in all but the multi-processor currently assigned to the linear system of interest. For a larger number of systems, the GPU results show a 1-2x speed-up over CPU NNLS due in part to the larger number of processing units suited for vector operations in the algorithm. The results between CPU NNLS and CPU lsqnonneg show the performance gains from fewer flops and memory accesses attained by the column reordering, updating, and downdating strategies.

5.7.3 Interaural Time Difference Estimation

For time-delay estimation, the publicly available CIPIC Head Related Transfer Function (HRTF) [1] consists of a collection of acoustic time-series measurements by microphones in various subject's left and right ears in response to direction-specific sound waves. The

[1] `https://lvis.gsfc.nasa.gov/index.php`

Number of systems	1	24	48	96	192
Number of updates	108	1477	2806	5695	12067
Number of downdates	14	220	406	834	1839
GPU NNLS	0.2172	0.2238	0.2356	0.4508	0.7203
CPU NNLS	0.0186	0.1636	0.2990	0.5989	1.2489
CPU lsqnonneg	0.0770	0.7163	1.2908	2.6225	5.5460
Matlab lsqnonneg	0.1342	1.1236	2.0152	4.1268	8.7176
Matlab FNNLS [108]	0.0493	0.5936	1.1135	2.2639	4.6148
Matlab interior-points [114]	8.3642	135.7896	261.3665	528.4468	1082.8714
Matlab PQN-NNLS [110]	1.4161	138.2953	217.6929	479.9867	862.9674

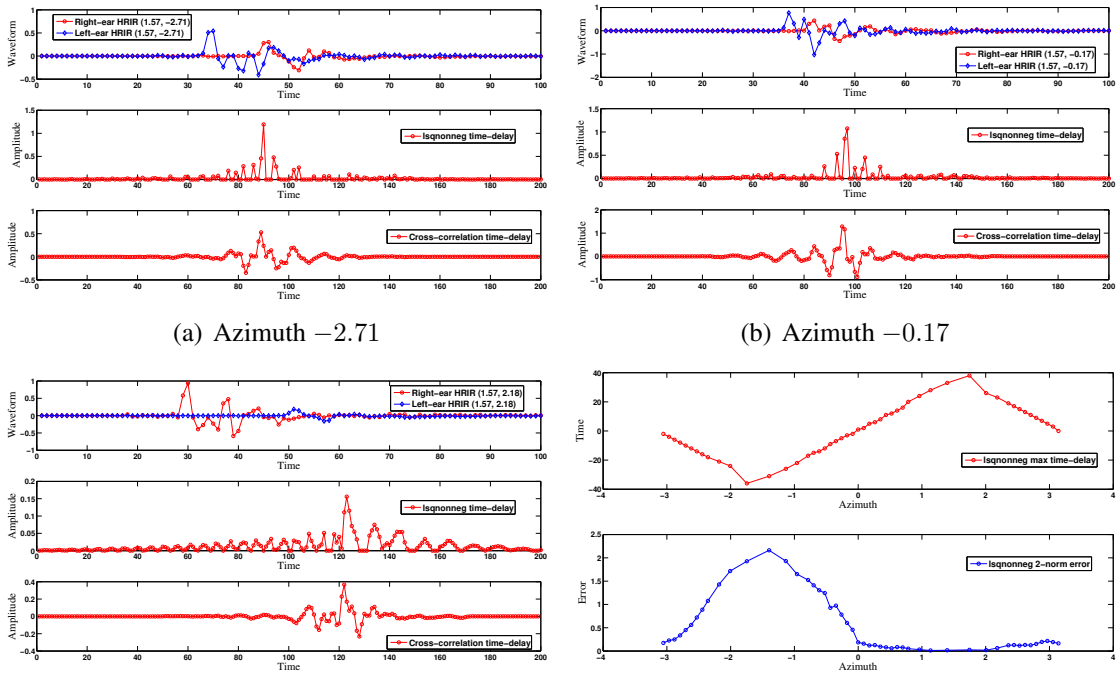
Table 5.4: Runtime (seconds) comparisons of NNLS and lsqnonneg variants for signal deconvolution. Signal and observation data taken from LVIS Sierra Nevada, USA (California, New Mexico), 2008.

time-series represents the scattering patterns of the sound source's acoustic wave off of the listener's anatomic features (torso, head, and outer ears) before reaching the eardrum. The frequency response of how sound is modified in phase and magnitude by such scattering is called the HRTF [7] and the time-series representation is called the head-related impulse response (HRIR). While analysis of the frequency representation is important for elevation cues, the left-right ITDs from the HRIR representation plays a more direct role to sound localization, especially along the azimuth ($\theta = \pi/2$) plane.

A CIPIC HRIR measurement consists of 100 samples over a 4.5 ms time-interval. Left-right ITDs for test subject 3 on the azimuth plane are shown Figure. 5.2. The NNLS time-delay formulation in Eq. 5.10, which can expressed by the convolution model in Eq. 5.9 by replacing functions s and x with s_r and s_l respectively, is compared to standard cross-correlation; the NNLS solution has a sparse representation that relates the left-ear HRIR to a non-negative linear combination of time-delayed right-ear HRIR. While cross-correlation may be used for time-delay estimation in microphone arrays whose minimum

group delay signals are identical, the left and right HRIR signals are sufficiently different due to sound scattering off of anthropometry. The NNLS time-delay formulation is also more interpretable than an unconstrained variant as the former solution is sparse and can be normalized to give a time-delay likelihood estimate.

The maximum valued ITDs are reported for each direction on the azimuth plane, exhibiting near 0 delay along directions close to the median ($\phi = 0$ and $\phi = \pi$) plane and the greatest differences along the orthogonal directions. The 2-norm reconstruction error $||Ax - b||$ is larger for source directions opposite the right ear, possibly due to a lower signal-to-noise ratio.



(a) Azimuth -2.71 (b) Azimuth -0.17

(c) Azimuth -2.16 (d) Max-delay and Error

Figure 5.2: ITD of left and right ear CIPIC HRIRs on the Azimuth plane for subject 3 are shown. The x-axis represents integer time-bins.

We also compare the NNLS solutions to that of unconstrained least squares, the cross-correlation, the time difference in maximum peak (15% energy) delays, and the

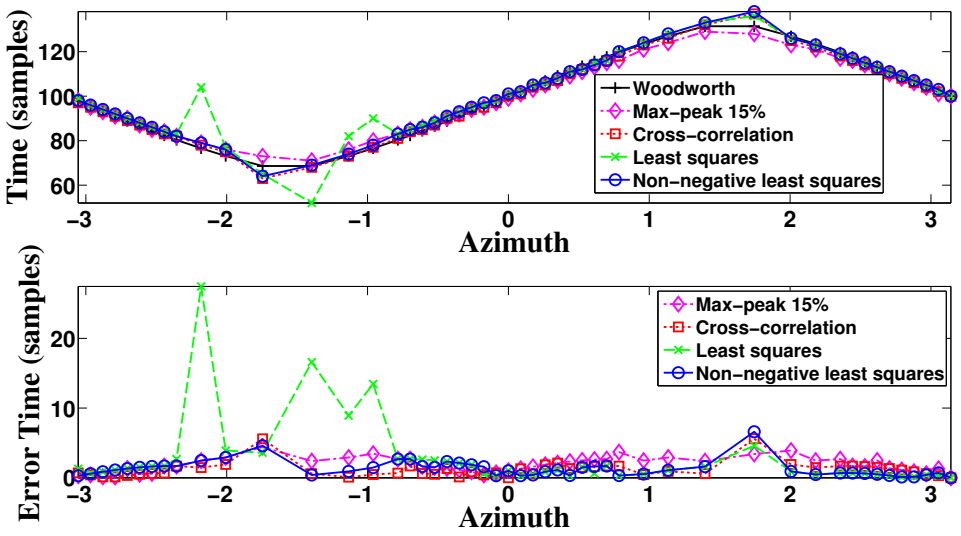


Figure 5.3: Horizontal plane ITD errors, computed over various methods (max-peak, cross-correlation, least squares, and NNLS), w.r.t. the Woodworth model [2] (ITD $= a(\phi + \sin\phi)/c$ for the sphere radius a anthropometric parameter $X_2/2$, sound speed c, and azimuth ϕ in radians) are shown.

theoretical Woodworth model for ITD on a rigid sphere [2]. NNLS enjoys several advantages: The solution is naturally sparse and optimal in the least-squares sense. The non-negative solution vector can be normalized to give a time-delay likelihood estimate. The largest weight in the solution encodes the maximum time-delay (treated as ITD) and is more distinct than cross-correlation (see Fig. 5.4). For a broader comparison, Fig. 5.3 compares the ITDs, for all horizontal plane directions, that are extracted using the listed methods. The maximum peak method in the unconstrained solution to Eq. 5.10 did not produce accurate ITDs compared to the NNLS solution when relating an IID attenuated right-HRIR s_r to the left-HRIR s_l on the negative azimuth side; it underestimates the time-delay along directions co-linear with the ears.

5.8 Conclusions

In this paper, we have presented an efficient procedure for solving least squares sub-problems in the active-set algorithm. We have shown that prior QR decompositions may be used to update and solve similar least squares sub-problems. Furthermore, a reordering of variables in the passive-set yielded fewer computations in the update step. This has lead to substantial speed-ups over existing methods in both the GPU and CPU implementations. Applications to satellite based terrain mapping, microphone array signal processing, and time-delay estimation for human sound localization are being worked on. Both GPU and CPU source codes are available on-line at `http://www.cs.umd.edu/~yluo1/Projects/NNLS.html`.

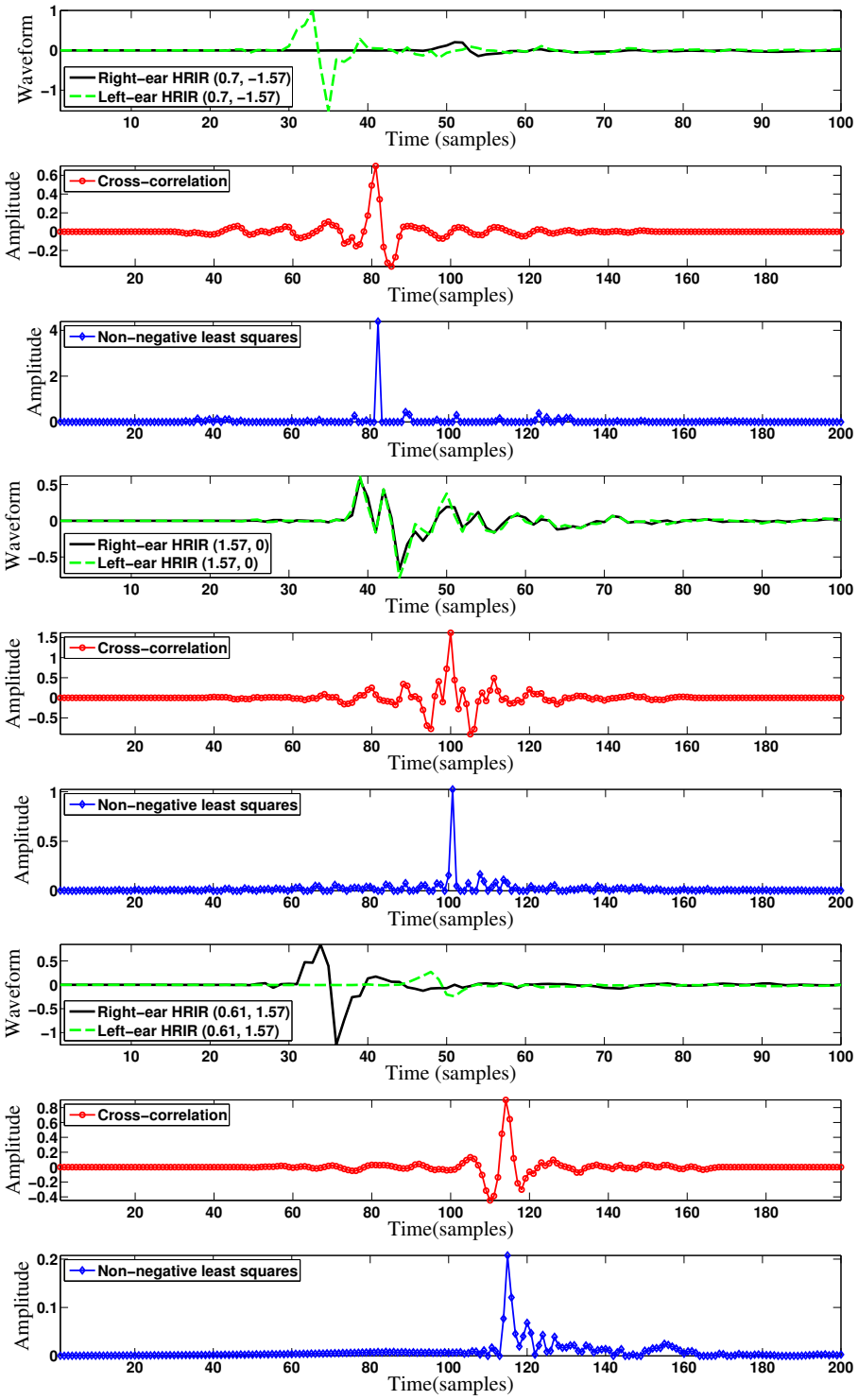


Figure 5.4: Cross-correlation and NNLS solutions for HRIR pairs on the azimuth plane (negative, zero, and positive time-delays centered at 100 time-samples) are shown.

Chapter 6: Sparse Head-Related Impulse Response for Efficient Direct Convolution

6.1 Introduction

The human sound localization ability is rooted in subconscious processing of spectral acoustic cues that arise due to sound scattering off the listener's own anatomy. Such scattering is quantified by a linear, time-invariant, direction-dependent filter known as the Head-Related Transfer Function (HRTF) [12]. HRTF knowledge allows presentation of realistic virtual audio sources in a Virtual Auditory Display (VAD) system so that the listener perceives the sound source as external to him/her and positioned at a specific location in space, even though the sound is actually delivered via headphones. A number of additional effects such as environmental modeling and motion tracking are commonly incorporated in VAD for realistic experience [13, 24].

The HRTF is typically measured by a placing a small microphone in an individual's ear canal and making a recording of a broadband test signal[1] emitted from a loudspeaker positioned sequentially at a number of points in space. The HRTF is the ratio of the spectra of microphone recording at the eardrum and at the head's center position in the absence

[1] Various test signals, such as impulse, white noise, ML sequence, Golay code, frequency sweep, or any broadband signal with sufficient energy in the frequencies of interest can be used for the measurements.

of the individual. Thus, the HRTF is independent of the test signal and the recording environment and describes the acoustic characteristics of the subject's anthropometry (head, torso, outer ears, and ear canal). The inverse Fourier transform of HRTF is the (time domain) filter's impulse response, called the Head-Related Impulse Response (HRIR).

The primary goal of the current work is to find a short and sparse HRIR representation so as to allow for computationally efficient, low latency time-domain convolution between arbitrary (long) source signal y and short HRIR x [126, 127]. It is expected that direct convolution[2] with short and sparse x would be more efficient w.r.t. latency and cost than frequency-domain convolution using the fast Fourier transform (FFT)[3] [44, 128].

Somewhat similar approaches has been explored in the literature previously. In the frequency domain, the HRTF has been decomposed into a product of a common transfer function (CTF) and a directional transfer function (DTF) [24, 36, 129], where the CTF is the minimum-phase filter with magnitude equal to average HRTF magnitude and the DTF is a residual. A more recent work on Pinna-Related Transfer Function (PRTF) [99, 130–132] provided successful PRTF synthesis model based on deconvolution of the overall response into *ear-resonance* (derived from the spectral envelope) and *ear-reflection* (derived from estimated spectral notches) parts. The novelty of the current work is that the *time-domain* modeling is considered and constraints are placed on "residual impulse response" (the time-domain analog of the DTF) to allow for fast and efficient real-time signal processing in time domain. Further, the tools to achieve this decomposition (semi-non-negative matrix factorization with Toeplitz constraints) are novel

[2] $(x * y)_i = \sum_j x_j y_{i-j+1}$ for x and y zero-padded as appropriate

[3] Fourier Transform convolution $x * y = \mathcal{F}^{-1}\{\mathcal{F}\{x\} \circ \mathcal{F}\{y\}\}$ for Fourier transform operator $\mathcal{F}\{\}$ and element-wise product $\circ$.

as well.

6.2 Problem Formulation

We propose the following time-domain representation of an HRIR $x \in \mathbb{R}^M$ given by

$$x \approx f * g, \quad g \geq 0, \tag{6.1}$$

where $*$ is the linear convolution operation, $f \in \mathbb{R}^{M-K+1}$ is a "common impulse response" derived from the subject's HRIR set, and $g \in \mathbb{R}^K$ is a sparse non-negative "residual"; the length of g is K. In analogy with terms commonly used in PRTF research, hereafter f is called the "resonance filter" and g the "reflection filter". The resonance filter is postulated to be independent of measurement direction (but of course is different for different subjects), and the directional variability is represented in g, which is proposed to represent instantaneous reflections of the source acoustic wave off the listener's anatomy; hence, g is non-negative and sparse. The computational advantage of such a representation is the ability to perform efficient convolution with an arbitrary source signal y via the associative and commutative properties of the convolution operation given by

$$y * x = (y * f) * g = (y * g) * f. \tag{6.2}$$

If y is known in advance, the convolution with f is direction-independent and can be precomputed in advance. Thereafter, direct time-domain convolution with a short and sparse g is fast and can be performed in real time. Moreover, even in the case of streaming

y, computational savings are possible if the output signal has to be computed for more than one direction (as it is normally the case in VAD for trajectory interpolation).

To learn the filters f and g, we propose a novel extension of the semi-non-negative matrix factorization (semi-NMF) method [49]. Semi-NMF factorizes a mixed-signed matrix $X \approx FG^T \in \mathbb{R}^{M \times N}$ into a product of a mixed-signed matrix F and a non-negative matrix G minimizing the approximation error in the least-squares sense. We modify the algorithm so that the matrix F has *Toeplitz structure*; then, FG^T is nothing but a convolution operation with multiple, time-shifted copies of f placed in columns of F (see Fig. 6.1). Thus, the overall approach for computing f and g is as follows: a) form matrix X from individual HRIRs, placing them as columns; b) run Toeplitz-constrained semi-NMF on X; c) take the first column and row of F as f; and d) for each direction, obtain non-negative g by taking a corresponding row of G.

The paper is organized as follows. In section 6.3, the modified semi-NMF algorithm is derived, with further extension to enforce a sparseness constraint on G by formulating it as a regularized L_1 norm non-negative least squares problem (L_1-NNLS) [59]. As the cost of time-domain convolution is proportional to the number of non-zero (NZ) elements in g, decreasing K (i.e., increasing sparsity) reduces computational load at the cost of increased approximation error. Experimental results are presented in section 6.4 along with the discussion. Finally, section 6.6 concludes the paper.

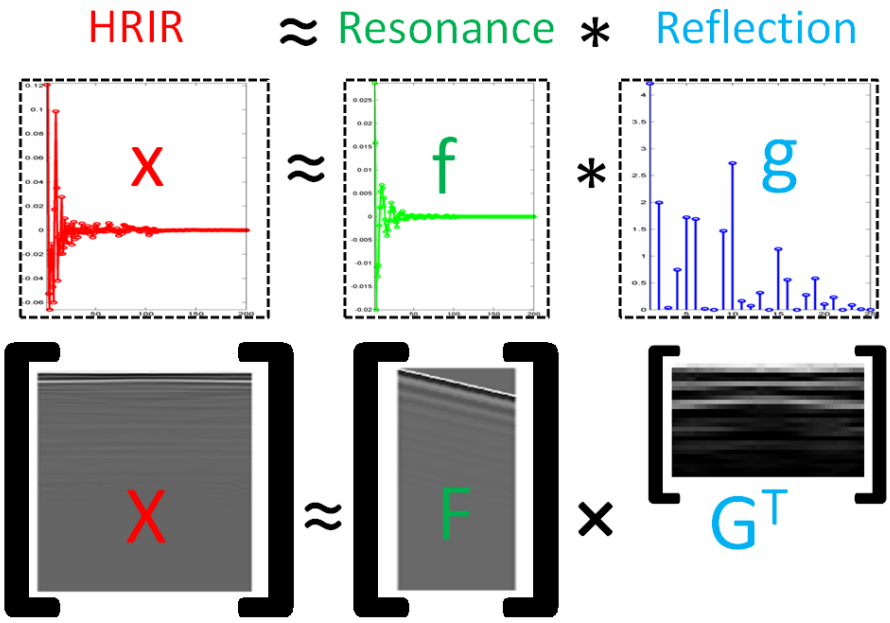


Figure 6.1: Modified semi-non-negative matrix factorization generalizes time-domain convolution for a collection of HRTFs X, resonance filter f, and non-negative reflection filters in G.

6.3 Semi-non-negative Toeplitz Matrix Factorization

6.3.1 Background

The original non-negative matrix factorization (NMF) [133] was introduced in the statistics and machine learning literature as a way to analyze a collection of non-negative inputs X in terms of non-negative matrices F and G where $X \approx FG^T$. The non-negativity constraints have been used to apply the factorization to derive novel algorithms for spectral clustering of multimedia data [134]. Semi-NMF [49] is a relaxation of the original NMF where the input matrix X and filter matrix F have mixed sign whereas the elements of G are constrained to be non-negative. Formally, the input matrix $X \in \mathbb{R}^{M \times N}$ is factorized into matrix $F \in \mathbb{R}^{M \times K}$ and matrix $G \in \mathbb{R}^{N \times K}$ by minimizing the residual Frobenius

norm cost function

$$\min_{F,G} \left\|X - FG^T\right\|_F^2 = \mathbf{tr}\left((X - FG^T)^T(X - FG^T)\right), \tag{6.3}$$

where $\mathbf{tr}\,()$ is the trace operator. For N samples in the data matrix X, the i^{th} sample is given by the M-dimensional row vector $X_i = X_{:,i}$ and is expressed as the matrix-vector product of F and the K-dimensional row vector $G_i = G_{i,:}$. The number of components K is selected beforehand or found via data exploration and is typically much smaller than the input dimension M. The matrices F and G are jointly trained using an iterative updating algorithm [49] that initializes a randomized G and performs an iterative loop computing

$$\begin{aligned} F &\leftarrow XG(G^TG)^{-1}, \\ G_{ij} &\leftarrow G_{ij}\sqrt{\frac{(X^TF)_{ij}^+ + [G(F^TF)^-]_{ij}}{(X^TF)_{ij}^- + [G(F^TF)^+]_{ij}}}, \\ (Q)_{ij}^+ &= \frac{|Q_{ij}| + Q_{ij}}{2}, \quad (Q)_{ij}^- = \frac{|Q_{ij}| - Q_{ij}}{2}. \end{aligned} \tag{6.4}$$

The positive definite matrix $G^TG \in \mathbb{R}^{K\times K}$ in Eq. 6.4 is small (fast to compute) and the entry-wise *multiplicative updates* for G ensure that it stays non-negative. The method converges to the optimal solution that satisfies *Karush-Kuhn-Tucker* conditions [49] as the update to G monotonically decrease the residual in the cost function in Eq. 6.3 for a fixed F, and the update to F gives the optimal solution for the same cost function for a fixed G.

6.3.2 Notational Conventions

To modify semi-NMF for learning the direction-independent f and a set of direction-dependent g, we introduce the following notation. Assume that $\tilde{F}$ is a Toeplitz-structured matrix and $\tilde{F}_{ij} = \Theta_{i-j}$ for parameters $\Theta = [\Theta_{1-M}, \ldots, \Theta_{K-1}]^T$; thus, all entries along diagonals and sub-diagonals of $\tilde{F}$ are constant. Hence, the Toeplitz structure is given by

$$\textbf{Top}\left(\Theta\right) = \begin{bmatrix} \Theta_0 & \Theta_1 & \ldots & \Theta_{K-2} & \Theta_{K-1} \\ \Theta_{-1} & \Theta_0 & \Theta_1 & \ldots & \Theta_{K-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \Theta_{2-M} & \ldots & \Theta_{-1} & \Theta_0 & \Theta_1 \\ \Theta_{1-M} & \Theta_{2-M} & \ldots & \Theta_{-1} & \Theta_0 \end{bmatrix}, \tag{6.5}$$

and is fully specified by parameters $\{\Theta_0, \ldots, \Theta_{K-1}\}$ and $\{\Theta_0, \ldots, \Theta_{1-M}\}$ along the first row and column. The Toeplitz matrix can also be represented indirectly as a linear combination of the parameters weighted by shift matrices $S^k \in \mathbb{R}^{M \times K}$ as

$$\tilde{F} = \sum_{k=1-M}^{K-1} S^k \Theta_k, \quad S^k_{ij} = \delta_{i,j-k}. \tag{6.6}$$

An arbitrary matrix F can be approximated by its nearest Toeplitz matrix $\tilde{F}$, which is defined as the minimizer of the residual Frobenius norm cost function given by

$$\begin{aligned} J &= \left\| F - \tilde{F} \right\|_F^2 = \textbf{tr}\left(F^T F - 2F^T \tilde{F} + \tilde{F}^T \bar{F} \right), \\ \frac{\partial J}{\partial \Theta_k} &= 2\textbf{tr}\left((F - \tilde{F})^T \frac{\partial \tilde{F}}{\partial \Theta_k} \right), \quad \frac{\partial \tilde{F}}{\partial \Theta_k} = S^k, \end{aligned} \tag{6.7}$$

where the partial derivatives of J w.r.t. Θ_k are linearly independent due to the trace term. By equating the derivatives to zero, the solution Θ is given by

$$\Theta_k = \frac{\mathbf{tr}\left(F^T S^k\right)}{\min(k+M, K-k, K, M)}. \tag{6.8}$$

Hence, a Toeplitz approximation $\tilde{F}$ to an arbitrary matrix F is obtained simply by taking the means of the subdiagonals of F.

6.3.3 Toeplitz-Constrained Semi-NMF

Assuming that a solution of the factorization problem F has in fact Toeplitz structure as per Eq. 6.6; the cost function in Eq. 6.3 is quadratic (convex) w.r.t. each Θ_k and the set of parameters Θ has a unique minimizer. The partial derivatives of the cost function[4] are given by

$$\begin{aligned} \frac{\partial \left\| X - \tilde{F} G^T \right\|_F^2}{\partial \Theta_k} &= \frac{\partial \mathbf{tr}\left((X - \tilde{F} G^T)^T (X - \tilde{F} G^T) \right)}{\partial \Theta_k} \\ &= 2\mathbf{tr}\left(\left(G^T G \sum_{i=1-K}^{M-1} S^{k^T} S^i \Theta_i \right) - S^{k^T} X G \right), \end{aligned} \tag{6.9}$$

where the product of shift matrices $S^{k^T} S^i$ can be expressed as the square shift matrix $\bar{S}^{i-k} = S^{k^T} S^i$. To solve for the set of parameters Θ, one needs to set the partial derivatives to zero, which yields a linear equation $A\Theta = b$ where $A \in \mathbb{R}^{|\Theta| \times |\Theta|}$, $|\Theta| = M + K - 1$ is

[4] Unlike the case considered in section 6.3.2, the partial derivatives in Eq. 6.9 are linearly dependent.

a Toeplitz square matrix, and $b \in \mathbb{R}^{M\times 1}$ is a vector specified as

$$A_{M+k,M+i} = \mathbf{tr}\left(G^T G \bar{S}^{i-k}\right), \quad b_{M+k} = \mathbf{tr}\left(S^{k^T} X G\right). \tag{6.10}$$

For positive-definite A, the matrix $\tilde{F}$ is given by the linear equation solution:

$$\tilde{F} = \mathbf{Top}\left(\Theta\right), \quad \Theta = A^{-1}b, \tag{6.11}$$

which is the unique minimizer of Eq. 6.3. Thus, to enforce Toeplitz structure on F, the iterative update $F \leftarrow XG(G^TG)^{-1}$ in the semi-NMF algorithm (Eq. 6.4) is replaced by computing F as prescribed by Eq. 6.10 and Eq. 6.11.

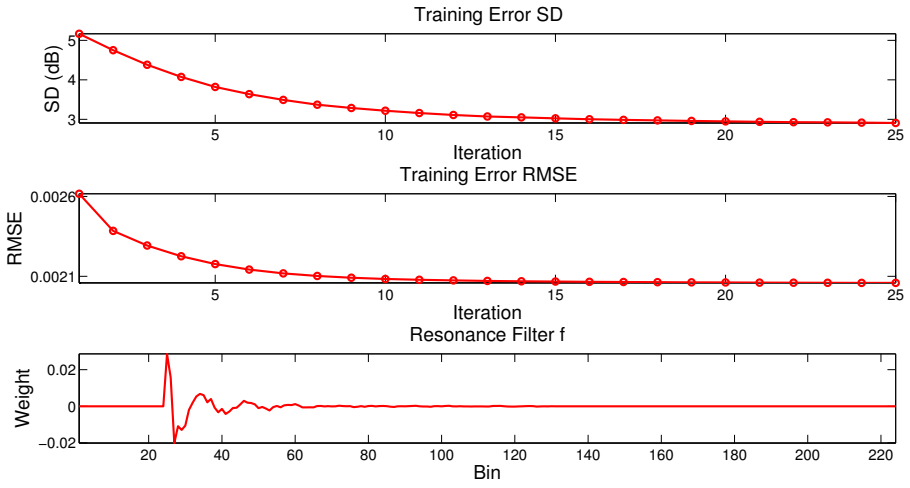


Figure 6.2: RMSE / SD error progress over 25 algorithm iterations.

Note that to perform a convolution between f and g (i.e., to reconstruct the HRIR) one needs to further constrain the Toeplitz matrix $\tilde{F}$ given in Eq. 6.5 in order to fulfill the filter length requirements. Such convolution is equal to the constrained Toeplitz matrix-

vector product

$$X_i = \begin{bmatrix} \Theta_0 & 0 & \dots & 0 \\ \Theta_{-1} & \Theta_0 & 0 & \dots \\ \vdots & \dots & \ddots & 0 \\ \Theta_{K-M} & \dots & \Theta_{-1} & \Theta_0 \\ 0 & \Theta_{K-M} & \dots & \Theta_{-1} \\ \vdots & \dots & \ddots & \vdots \\ 0 & \dots & 0 & \Theta_{K-M} \end{bmatrix} \begin{bmatrix} G_{i1} \\ \vdots \\ G_{iK} \end{bmatrix}, \tag{6.12}$$

where the parameters $\{\Theta_{K-M-1}, \dots, \Theta_{1-M}, \Theta_1, \dots, \Theta_K\}$ are set to zero. Only the NZ parameters $\{\Theta_0, \dots \Theta_{K-M}\}$ are solved for in a smaller $(M-K+1) \times (M-K+1)$ sized linear system as per Eq. 6.10 and Eq. 6.11. These NZ parameters form the resonance filter f:

$$f = \{\Theta_0, \dots \Theta_{K-M}\} \in \mathbb{R}^{M-K+1}. \tag{6.13}$$

6.3.4 Minimizing the Number of Reflections

To introduce sparsity, we restrict the number of NZ entries (NNZE) in G. In order to do that, we fix the trained resonance filter $\tilde{F}$ and solve for each reflection filter $g = G_i$ separately in a penalized L_1-NNLS problem formulation [114] given by

$$\min_{G_i} \left\| \mathcal{D}\left(FG_i^T - X_i\right) \right\|_2^2 + \lambda \left|G_i\right|_1, \quad \text{s.t. } G_i \geq 0, \tag{6.14}$$

where $D \in \mathcal{R}^{M_* \times M}$ is some transformation of the residual[5]. Three transformations are considered.

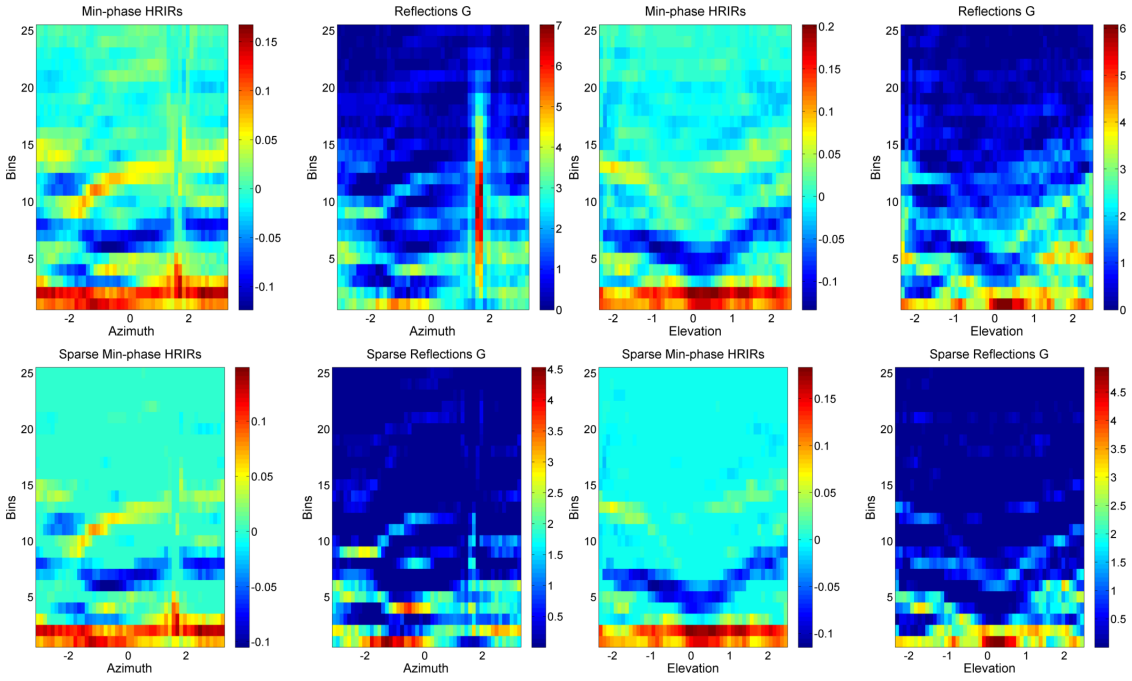


Figure 6.3: Top row: Slices of reflection filter matrix G trained without sparsity constraint; also, original HRIR after min-phase processing, time delay removing, and normalization. Bottow row: Slices of reflection filter matrix G trained with sparsity constraint applied ($\lambda = 10^{-3}$); also, HRTR reconstructed from it.

1. The identity transform $\mathcal{D}_I = I \in \mathbb{R}^{M \times M}$, which directly minimizes the residual norm while penalizing large magnitudes in the reflection filter G_i.

2. The convolution transform

$$\begin{aligned} \mathcal{D}_C &= \mathbf{Top}\left(\Theta^C\right) \in \mathbb{R}^{M \times M}, \\ \Theta^C_{1:M-1} &= \mathcal{N}_\sigma(1 : M-1), \quad \Theta^C_{0:1-M} = \mathcal{N}_\sigma(0 : 1-M), \end{aligned} \tag{6.15}$$

[5]A free Matlab solver for L_1-NNLS is available online at `http://www.stanford.edu/~boyd/papers/l1_ls.html`

which is characterized by the Gaussian filter $\mathcal{N}_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$. This transform effectively low-passes the reconstructed HRIR. It is equivalent[6] to windowing the frequency-domain residuals with a Gaussian filter of inverse bandwidth; hence, the low-frequency bins are weighted heavier in the reconstruction error.

3. The window transform

$$\mathcal{D}_W = \mathbf{diag}\left(v_\sigma(0 : M-1)\right) \in \mathbb{R}^{M\times M}, \tag{6.16}$$

where $v_\sigma(x) = e^{-\frac{x^2}{\sigma^2}}$ is a Gaussian-like filter. The window transform has the effect of convolving the signal spectrum with a filter $v_\sigma(x)$ as if both were time series, which is equivalent to windowing HRIR in time domain by the Gaussian filter of inverse bandwidth. In this way, the earlier parts of the reconstructed HRIR contribute to the reconstruction error to the larger extent.

The additional regularization term λ in Eq. 6.14 affects the sparsity of g as increasing λ decreases the NNZE. In our practical implementation, we also discard elements that are technically non-zero but have small ($\leq 10^{-4}$ magnitude) as they contribute little to the reconstruction. The final algorithm for learning the resonance and reflection filters with the sparsity constraint on the latter is summarized in Algorithm 13.

[6]Convolution in time domain is equivalent to windowing in frequency domain, and vice versa.

Algorithm 13 Modified Semi-NMF for Toeplitz Constraints

Require: Filter length K, transformation matrix $D \in \mathbb{R}^{M*\times M}$, HRIR matrix $X \in \mathbb{R}^{M\times N}$, max-iterations T

$G \leftarrow$ **rand**(N, K) \\ Random initialization
for $t = 1$ to T **do**
  $\Theta \leftarrow A^{-1}b$ \\ Solve for resonance via Eqs. 6.10, 6.11
  $\tilde{F} \leftarrow$ **Top** (Θ) \\ Toeplitz matrix via Eqs. 6.12, 6.13
  Update G. \\ Multiplicative update via Eq. 6.4
end for
Fine-tune G. \\ Vary λ, σ in Eqs. 6.14, 6.16, 6.15
return $\tilde{F}$, G

6.4 Experiments and Results

6.4.1 HRIR/HRTF Data Information

We have performed an extensive series of experiments on the data from the the well-known CIPIC database [1]; however, the approach can be used with arbitrary HRTF data [25, 26, 135, 136]. We pre-process the data as follows: a) convert HRIR to min-phase; b) remove the initial time delay so that the onset is at time zero; and c) normalize each HRIR so that the absolute sum over all samples is equal to unity.

As mentioned previously, our processing intends to separate the arbitrary impulse response collection of into "resonance" (direction-independent) and "reflective" (direction-dependent) parts. For the HRIR, we believe that these may correspond to pinna/head resonances and instantaneous reflections off the listener's anthropometry, respectively. Such an approach may also be applicable to other IR collections; for example, room impulse responses [137] may be modeled as a convolution between a shared "resonance" filter (i.e. long reverberation tail) and the "reflective" filter (early sound reflections off the walls). In order to obtain a unique decomposition using Algorithm 13, one would need to have

the number of directional IR measurements larger than the IR filter length, which may be impractical. This topic is a subject of future research.

6.4.2 Error Metric

For evaluation, we consider two error metrics – the root-mean square error (RMSE) and the spectral distortion (SD), representing time-domain and frequency-domain distortions respectively:

$$\begin{aligned} \text{RMSE} &= \sqrt{\frac{\left\| \left(X - \tilde{F} G^T \right) \right\|_F^2}{MN}}, \\ \text{SD}\left(H^{\{j\}}, \tilde{H}^{\{j\}} \right) &= \sqrt{\frac{1}{M} \sum_{i=1}^{M} \left(20 \log_{10} \frac{|H_i^{\{j\}}|}{|\tilde{H}_i^{\{j\}}|} \right)^2}, \end{aligned} \tag{6.17}$$

where X_j is the reference HRIR, $\tilde{F} G_j^T$ is the reconstruction of it, $H^{\{j\}} = \mathcal{F}\{X_j\}$ is the reference HRTF, X_j is the reference HRIR, and $\tilde{H}^{\{j\}} = \mathcal{F}\left\{ \tilde{F} G_j^T \right\}$ is the HRTF reconstruction.

Another feasible comparison is validation of the reconstruction derived from sparse representation (Eq. 6.14) against the naive regularized least squares (L_1-LS) approximation of HRIR X_i given by

$$\min_{\hat{x}} \left\| \mathcal{D}\left(\hat{x} - X_i \right) \right\|_2^2 + \lambda \left| \hat{x} \right|_1, \tag{6.18}$$

where $\hat{x} \in \mathbb{R}^{M \times 1}$ (i.e. magnitude-constrained approximation without non-negativity constraint). The difference between SD error of L_1-NNLS approximation and of L_1-LS ap-

proximation is a metric of advantage provided by our algorithm in comparison with LS HRIR representation, which retains large-magnitude HRIR components irrespective of their sign.

6.4.3 Resonance and Reflection Filter Training

The resonance and reflection filters f and G are jointly trained via Algorithm 13 for 50 iterations for $N = 1250$ number of samples, $M = 200$ time-bins, and $K = 25$ filter length using left-ear data of CIPIC database subject 003. N and M here are fixed (they are simply the parameters of the input dataset). The choice of K is somewhat arbitrary and should be determined experimentally to obtain the best compromise between computational load and reconstruction quality. Here we set it to the average human head diameter ($\approx$ 19.2 cm) at the HRIR sampling frequency (44100 Hz). Visual HRIR examination reveals that most of the signal energy is indeed concentrated in the first 25 signal taps.

Fig. 6.2 shows RMSE and SD error over 50 iterations of Algorithm 13 with no sparsity constraint on G (i.e. $\lambda = 0.0$). The final filter f is a periodic, decaying functions resembling a typical HRIR plot. The final matrix G is shown in the top row of Fig. 6.3. The mean NNZE for G is 22.74 (it is less than K due to removal of all elements with magnitude less than 10^{-4}). As it can be seen, the SD error achieved is 3.0 dB over the whole set of directions.

In order to obtain the sparse HRIR representation, we re-ran the algorithm using identity transformation in L_1-NNLS constraint and a fixed $\lambda = 10^{-3}$ (this parameter was determined empirically to cut the NNZE approximately in half). The final matrix G

obtained in this case is shown in the bottom row of Fig. 6.3. It is sparse as expected and has a number of non-zero bands spanning the time-direction domain; thus, only the most salient components of G are retained. In this case, the mean NNZE is 11.48 and the SD error is 5.3 dB over the whole set of directions. In the following section, the guidelines for setting λ are considered.

6.4.4 Regularization Term Influence

We investigate the effects of varying the λ term in Eq. 6.14 under the identity transform $\mathcal{D}_I$ on the NNZE in G and on the RMSE / SD error. A sample HRIR is chosen randomly from the data set. Fig. 6.4 shows the effect of changing λ on NNZE, RMSE, SD error, and reconstructed HRIR/HRTF *per se*. The trends that one can see in the figure are consistent with expectation; it is interesting to note that as λ increases, low-magnitude elements in G are discarded whereas both the dominant time-domain excitations and the shape of the spectral envelope in the reconstructed HRIR are preserved.

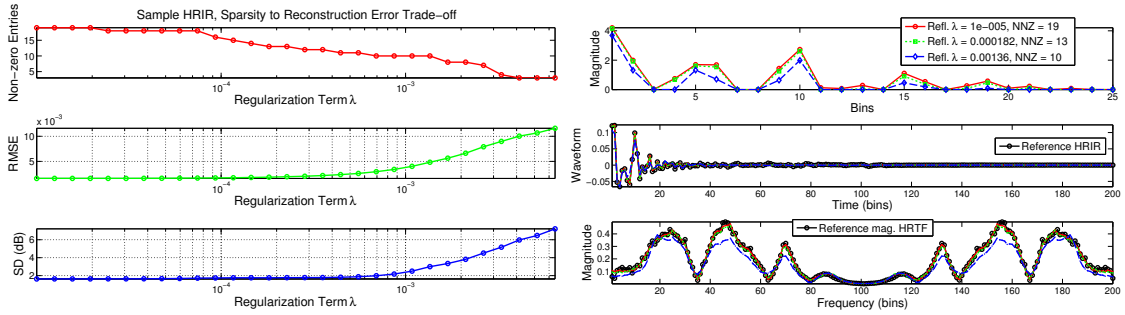


Figure 6.4: Influence of the L_1 regularization term λ in Eq 6.14 on NNZE and on the reconstruction error for sample HRIR.

Further analysis of the NNZE and of the SD error over the full set of HRIR measurement directions is shown in Fig. 6.5. Note that ipsilateral reflection filters have lower

NNZE[7] and achieve lower SD error. This is understandable, as they do fit better into a "resonance-plus-reflections" model implied in this work. On the other hand, contralateral HRIR reconstruction requires larger NNZE and results in more distortion, presumably due to significant reflections occuring later than $K = 25$ time samples; note that while some effects of head shadowing (attenuation / time delay) are removed in the preprocessing step, others may not be modeled accurately; on the other hand, accurate HRIR reproduction on contralateral side is not believed to be perceptually important [138]. Improvement in quality of contralateral HRIR reconstruction is a subject of future research. One approach is to learn separate HRIR decomposition, possibly with different length of f / g filters, for different sub-regions of space.

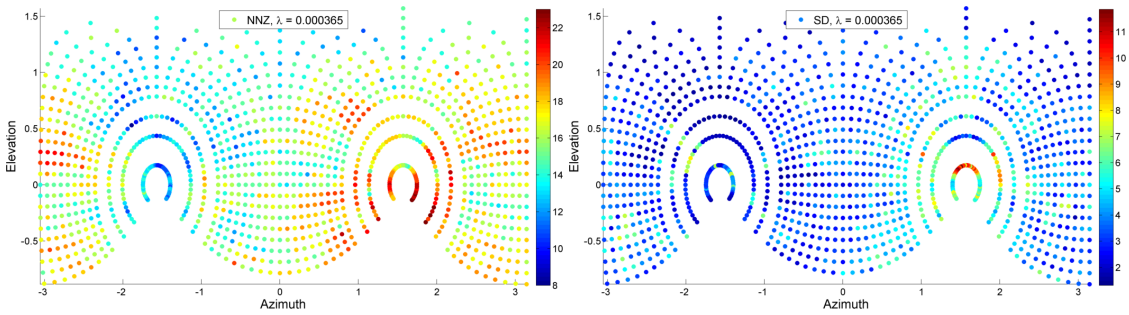


Figure 6.5: A map of NNZE and SD error over the full spherical coordinate range for left-ear HRIR data. Note smaller NNZE / SD values on ipsilateral side.

Finally, in Fig. 6.6 we compare the L_1-NNLS reconstruction against the naive L_1-LS reconstruction in terms of the convolution filter NNZE and SD error for varying λ and a number of directions selected on horizontal and on medial planes. For all of these, the difference between solutions is less than 2.0 dB SD; further, for 13 (out of 16) cases the L_1-NNLS solution has the same or better reconstruction error than naive L_1-LS solution

[7]The variability exhibited can not be due simply to total HRIR energy differences as they were all normalized during pre-processing.

in highly-sparse (NNZE $\leq K/2$) case. This implies that our decomposition is able to find a resonance filter and a sparse set of early reflections that represent the HRTF better than the dominant magnitude components of the original HRIR *per se*.

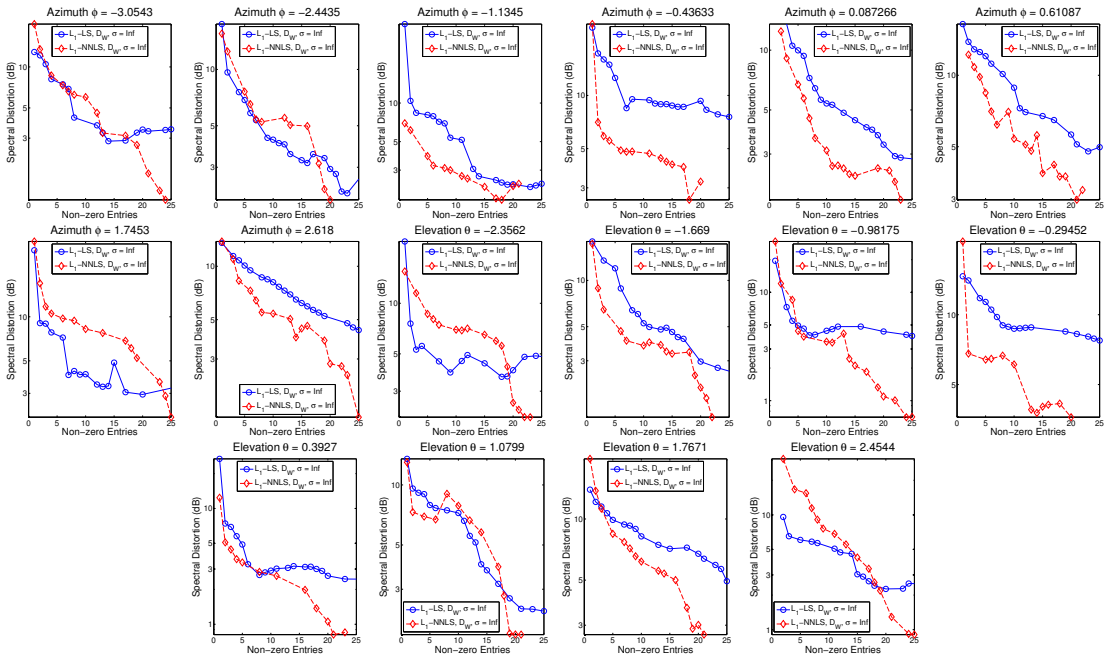


Figure 6.6: A comparison between varying-sparsity L_1-NNLS and L_1-LS solutions for selected directions on horizontal and median planes. Angles are listed in radians.

6.4.5 Transformation Bandwidth Optimization

Further reduction of the SD error is possible via use of transform functions defined in section 6.3.4. Application of these functions would result in different weights placed on different aspects of reconstructed HRIR. Hence, we investigate the selection of bandwidth term σ in Eq. 6.16 with no L_1 penalty term ($\lambda = 0$) for the window transform[8].

As mentioned before, application of the window transform $\mathcal{D}_W$ causes smoothing in the frequency domain; the amount of smoothing depends on the bandwidth term σ. Fig.

[8]We omit the convolution transform $\mathcal{D}_C$ in experiments as applying a low-pass filter to the residuals entails a per-frequency error metric.

Table 6.1: Mean spectral distortion for individually tuned $\mathcal{D}_{W,\sigma}$

	H-plane	**M-plane**	**All directions**
$\sigma \to \infty$	2.72	1.73	2.49
Tuned σ	2.53	1.57	2.24

6.7 shows the SD error dependence on σ for one sample HRIR. Obviously as bandwidth $\sigma \to \infty$, the window transform becomes the identity transform; indeed, SD error stays constant for $\sigma > 70$. It can be seen though that the minimum SD error occurs at a finite $\sigma = 30$ (for this particular HRIR). The parameter σ can be efficiently fine-tuned (via fast search methods) *separately* for each HRIR in the subject's HRTF set. Table 6.1 compares the SD error obtained over the grid of $\sigma = [15+((0:24)*2), 100, 160, 250]$ using window transform to the SD error with identity transform (which is the same as window transform with $\sigma \to \infty$) across horizontal / median plane and over all HRTF set directions. It can be seen that on average, such tuning decreases the SD error by about 10%.

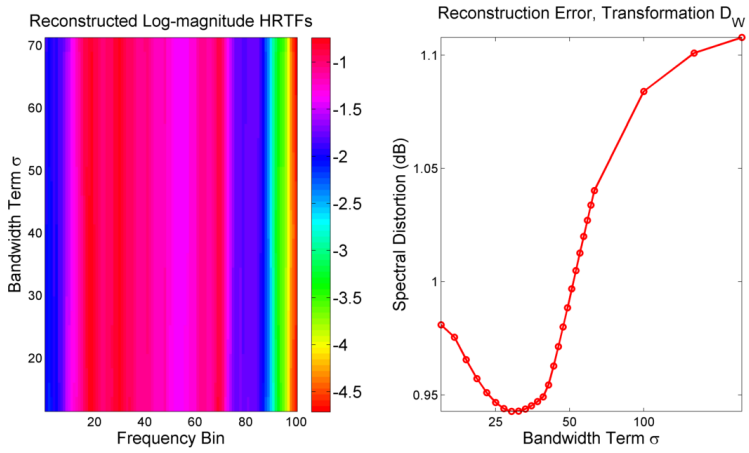


Figure 6.7: SD error dependence on bandwidth of window transform for a sample HRIR.

6.4.6 Computational Cost

Consider the cost of computing the i^{th} sample of $(x * y)_i$ where $*$ is the convolution operation. Direct time-domain convolution requires $\min\{|x|, |y|\}$ real floating-point operations, where $|x|, |y|$ is the NNZE in each filter. In practice, convolution is normally done in blocks of fixed size (so-called partitioned convolution). In case of time-domain processing, partitioned convolution incurs neither memory overhead nor latency.

At the same time, the state-of-the-art frequency-domain implementation [139] requires $\frac{68}{9}(|y| \log_2 |y| + |y|)/(|y| - |x| + 1)$ complex floating-point operations per output sample. For a long input signal (e.g. $|y| = 44100$ – i.e. one second at CD audio quality), time-domain algorithm is faster than frequency-domain implementation for $|x| < 127$. Further, in real-time processing, latency becomes an issue, and one must use partitioned convolution (with reasonably small block size) and the *overlap-and-save* algorithm [35]. In order to achieve e.g. 50 ms latency, one must have $|y| = 2205$. For this segment length, direct time-domain convolution incurs less computational cost when $|x| < 90$. Thus, a time-domain convolution using sparse filter x as derived in this paper is arguably quite beneficial to the computational load incurred by the VAD engine.

6.5 Discussion

While our study presents the theoretical derivation of our factorization algorithm, a number of practical concerns have been omitted for reasons of scope. We provide a number of remarks on these below.

First, an optimal NNZE is hardware dependent, as the crossover point between

time-domain and frequency-domain convolution costs depends on the computational platform as well as on the specific implementations of both. For example, specialized digital signal processors can perform efficient real time-domain convolution via hardware delay lines whereas being less optimized for handling complex floating-point operations necessary for fast Fourier transform.

Second, the target reconstruction error can be adjusted to match a desired fidelity of spatialization. For instance, early reflections off nearby environmental features may have to be spatialized more distinctly than a number of low-magnitude later reflections that collectively form the reverberation tail. Further, the need to individually optimize the penalty term λ for each direction depends also on desired sparsity (i.e. computational load) versus SD error trade-off. Such real-time load balancing is an open challenge that depends on available computational resources on specific hardware platform.

Certain obvious extensions of the work presented has also not been fully described for clarity. We note that using non-zero λ term and varying the bandwidth σ in $\mathcal{D}_W$, $\mathcal{D}_C$ transforms could lead to decrease in SD error at the same NNZE when tuned. A set of bandpass transformations that constitute the orthogonal basis for the discrete Fourier transform could also be used, as in this case the error could be weighted individually in each frequency band to match the listener's characteristics (e.g. by using the equal loudness contours in frequency).

Another consideration is the choice of the cost function in Eq. 6.3, which currently omits prior information on the HRIR measurement direction distribution. It may be undesirable to place equal weight on all directions if those are in fact spaced non-uniformly. Instead, the sample residual can be biased by introducing a kernel transforma-

tion $\mathcal{D} \in \mathbb{R}^{N \times N}$ of the HRIR measurement directions ($\mathcal{D}_{ij}$ is a kernel function evaluation between directions i^{th} and j^{th}) into the cost function $\mathbf{tr}\left((X - FG^T)\mathcal{D}^{-1}(X - FG^T)^T\right)$, which would decorrelate HRIR reconstruction error in densely-sampled area and thus avoid giving preferential treatment to these areas while optimizing.

6.6 Conclusions

We have presented a modified semi-NMF matrix factorization algorithm for Toeplitz constrained matrices. The factorization represent each HRIR in a collection as a convolution between a common "resonance filter" and specific "reflection filter". The resonance filter has mixed sign, is direction-independent, and is of length comparable to original HRIR length. The reflection filter is non-negative, direction-dependent, short, and sparse. The tradeoff between sparsity and approximation error can be tuned via the regularization parameter of L_1-NNLS solver, which also has the ability to place different weights on errors in different frequency bands (for HRTF) or at different time instants (for HRIR). Comparison between HRIR reconstructed using the proposed algorithm and L_1-LS reference solution shows that the former has much better sparsity-to-error tradeoff, thus allowing for high-fidelity latency-free spatial sound presentation at very low computational cost.

Chapter 7: Conclusions

This thesis developed several novel solutions for fast spatial audio rendering and personalization via numerical and machine learning methods. First, we developed an HRTF based sound-source localization model using two receivers. Binaural input features consisting of ratios between same-direction left and right ear HRTFs were extracted; GP-SSL models, trained on known binaural inputs, were used to predict sound-source directions. Next, we introduced an active-learning problem for inferring HRTFs in listening tests. The GP-SSL models were extended for the prediction of SSL errors from the same inputs. This was used to solve the query-selection problem for recommending HRTFs to the listener for localization. Experiments showed that the recommended HRTFs achieved smaller localization errors by both human and GP-SSL virtual listeners than the initial non-individualized HRTF guesses.

Next, we developed a novel tensor product formulation for GPR and sparse-GPR covariance matrices. The formulation exploited the gridded structure between the gridded input domains and enables the efficient factorization of large Gram matrices via the Kronecker product decomposition. The model was adapted for the fast interpolation of HRTFs over the joint spherical coordinate and frequency domains. This also solved the problem of fusing multiple HRTF datasets (same-subject, different labs) and learning a

series of data transformations which explained the inter-lab dataset variances. Experimental results showed that GP models had lower generalization error than other spherical interpolation models.

Last, we showed that collections of HRIRs can be decomposed into a long direction-independent filter and short/sparse direction-dependent filters by constraining a non-negative matrix factorization algorithm to Toeplitz structured matrices. This reduced the computational costs of time-domain convolution with arbitrary input sound-sources and proven to be faster than frequency-domain convolution via FFT. The direction-dependent filters can be sparsified by solving a penalized NNLS problem; we developed a low-rank updating NNLS algorithm and parallelized it for multi-core (CPU/GPU) processors.

7.1 Open Problems

7.1.1 Toeplitz Matrix Factorizations for Blind-Dereverberation

Single-source blind-dereverberation is an important problem for removing the effects of both long and short time-delayed reflections of an arbitrary input signal x off a complex environment. Under LTI system assumptions, the observed signal x can be expressed as the convolution between the environment's transfer function (filter f) and a "clean" source-signal (filter g). The effects of reverberation in the environment often reduces the intelligibility of x; recovering g is the frequency domain is easy if f is already known/measured.

However in the case where both f and g are unknown, the problem is underdetermined due to the large number of unknowns (filter weights of f and g); prior assumptions

about the distribution of f and g must be added as constraints. Consider the time-domain convolution $x = f * g$, formalized in terms of a double Toeplitz matrix factorization by concatenating windowed segments of x (length N). This matrix system is given by

$$X \approx FG, \quad X = [x_{1:K}, x_{2:(K+1)}, \ldots, x_{(N-K+1):N}], \tag{7.1}$$

where F and G are Toeplitz structured matrices with first row and columns defined by respective zero-padded filters f and g (lengths K and $N - K + 1$). Solving for $F|G$ or $G|F$ could follow Eqs. 6.10, 6.11 presented in chapter 6. However, neither f or g are constrained to be non-negative and so the locally converged solution will lack interpretation; the "correctness" of the solutions will be difficult to evaluate as the minimizers of the least squares reconstruction error in Eq. 6.3 are not indicative of a probable f and an intelligible g.

The open problem concerns how to encode prior knowledge on parameters f, g without rendering the filter learning algorithm intractable. The Toeplitz matrix formulation in Eq. 7.1 may be useful with regards to structuring the learning algorithm if variants of expectation-maximization [51] are used. Establishing the priors on f and g is more difficult so we suggest several approaches. Priors on f may be probabilistically modeled from existing reverb filters collected from common-place room and outdoor environments. Priors on g are more varied as intelligibility measures are domain specific (e.g. speech, music instrument, animals); incorporating expert-domain knowledge is expensive so we suggest learning generative models such as deep restricted Boltzmann machines [140] on available datasets. Low-entropy assumptions such as a sparse g are also

valid constraints to consider.

7.1.2 Alternative Covariance Functions for Gaussian Processes

The design of the GP covariance function represents the prior domain assumptions regarding how similarities in the observations are explained by similarities in inputs. For GP HRTF interpolation models (chapter 3, it is possible to improve the data LMH (goodness-of-fit) by considering covariance functions that may violate the gridded and separable assumptions between spherical coordinate and frequency input domains. The resulting Gram matrix K may not have a Kronecker product decomposition and would require fast approximation methods such as Conjugate gradient [141] and Nyström approximation [77] to solve the linear system of equations in Eq. 3.5 in a tractable manner. Several possibilities are given: Covariance functions for the spatial domain may benefit from non-stationarity [142] as the smoothness of HRTFs may vary along directions that are shadowed by the head. Non-stationarity in the frequency domain may represent changing smoothness due to sound reflections that would only occur within restricted frequency ranges correlated with the sizes of anthropometry features.

For HRTF based GP-SSL models from chapter 2, the product of independent Matérn class covariance functions over frequency bins in Eq. 2.7 can be formulated as a single covariance function with a diagonal-covariance matrix that represents the bandwidth hyperparameters. This can be generalized by a full-covariance term which would account for cross-covariances between frequencies and increase the expressibility of the models.

For example, the modified squared exponential kernels can be expressed by

$$K(x_i, x_j) = e^{-(x_i - x_j)^T A^{-1}(x_i - x_j)}, \quad A^{-1} = LDL^T, \tag{7.2}$$

where A^{-1} is expressed in the form of a Cholesky decomposition (product of a lower triangular matrix L, a diagonal matrix D, and the transpose L^T); matrix entries in the decomposition are treated as the hyperparameters to be optimized under the data LMH criterion via Eq. 2.8. The training time is expected to increase due to the larger number of hyperparameters.

7.1.3 Perceptual Measures of HRTF Similarity

While spectral distances such as SD (Eq. 6.17 and the Itakura-Saito distance [143] are typically used in speech intelligibility tests, they may not be suited for all domains in the field of acoustics. A trivial example evaluates the distance between two pure tones where their SD would be independent of how far they are separated in frequency. One alternative measure is the earth mover's distance (EMD) [144] which measures the dissimilarity between two probability distributions in terms of the minimum *work* for moving masses over a distance so that the distributions match. EMD distances have been used to compare histograms of color statistics (profiles) for image retrieval; the analogous histogram concept for acoustic waveforms is its magnitude frequencies (spectral energy) which can be normalized to sum to unity. Moreover, the EMD distance between non-negative D dimensional unit vectors can be efficiently computed via the sum of absolute differences

between their cumulative distribution functions given by

$$\text{EMD}(x, y) = \sum_{i=1}^{D} \left| \bar{x}_i - \bar{y}_i \right|, \quad \bar{x}_i = \sum_{j=1}^{i} x_j. \tag{7.3}$$

This may be useful for magnitude spectra such as HRTFs which can be characterized by frequency-dependent extrema values (peaks and notches) and correlated with physical anthropometry features.

Another problem concerns the methodology for comparing two HRTFs. Perceptual evaluations of HRTFs are naturally subjective as they relate to a measurement direction to the subject's response within his/her ear canal. While we have shown that HRTFs can be localized via listening tests by both humans and GP-SSL models (chapter 2), their localization directions are unique due filtering w.r.t. to his/her own set of HRTFs. It may be of future interests to perform studies (collect statistics) on how other individual's HRTFs are mismatched in their reported directions by the human population or via the GP-SSL models. Thus, future perceptual distances between HRTFs can be conceived by either their physical localization distances in the former or statistical distances between probability distributions in the latter.

Bibliography

[1] V. R. Algazi, R. O. Duda, and C. Avendano, "The CIPIC HRTF Database," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, 2001, pp. 99–102.

[2] R. Woodworth and G. Schlosberg, *Experimental psychology*. Holt, Rinehard and Winston, 1962.

[3] F. Rumsey, *Spatial audio*. Taylor & Francis, 2001.

[4] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 749–752.

[5] J. G. Beerends and J. A. Stemerdink, "A perceptual audio quality measure based on a psychoacoustic sound representation," *Journal of the Audio Engineering Society*, vol. 40, no. 12, pp. 963–978, 1992.

[6] S. A. Gelfand and H. Levitt, *Hearing: An introduction to psychological and physiological acoustics*. Marcel Dekker New York, 1998.

[7] J. Blauert, *Spatial hearing: the psychophysics of human sound localization*. Cambridge, Massachusettes: MIT Press, 1997.

[8] "10 trends that will shape consumer mindset and behavior in 2013," JWT Intelligence, accessed: 2012-12-04.

[9] L. Rayleigh, "On our perception of the direotion of a source of sound," *Proceedings of the Musical Association*, vol. 2, no. 1, pp. 75–84, 1875.

[10] R. O. Duda, C. Avendano, and V. R. Algazi, "An adaptable ellipsoidal head model for the interaural time difference," in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, vol. 2. IEEE, 1999, pp. 965–968.

[11] D. Wright, J. H. Hebrank, and B. Wilson, "Pinna reflections as cues for localization," *The Journal of the Acoustical Society of America*, vol. 56, no. 3, pp. 957–962, 1974.

[12] D. R. Begault, "3D sound for virtual reality and multimedia," *Academic Press, Cambridge, MA*, 1994.

[13] D. Zotkin, R. Duraiswami, and L. S. Davis, "Rendering localized spatial audio in a virtual auditory space," *IEEE Transactions on Multimedia*, vol. 6, pp. 553–564, 2004.

[14] R. Duraiswami, D. N. Zotkin, Z. Li, E. Grassi, N. A. Gumerov, and L. S. Davis, "High order spatial audio capture and binaural head-tracked playback over headphones with hrtf cues," 2005.

[15] Y. Luo and R. Duraiswami, "Efficient parallel non-negative least squares on multicore architectures," *SIAM Journal of Scientific Computing*, 2011.

[16] Y. Luo, D. N. Zotkin, H. Daumé III, and R. Duraiswami, "Kernel regression for head-related transfer function interpolation and spectral extrema extraction," in *ICASSP*, 2013.

[17] Y. Luo and R. Duraiswami, "Fast near-GRID Gaussian process regression," *AISTATS*, vol. 31, pp. 424–432, 2013.

[18] Y. Luo, D. N. Zotkin, and R. Duraiswami, "Statistical analysis of head related transfer function (HRTF) data," in *International Congress on Acoustics*, 2013.

[19] Y. Luo, D. N. Zotkin, and R. Duraiswami, "Gaussian process data fusion for heterogeneous HRTF datasets," in *WASPAA*, 2013.

[20] Y. Luo, D. N. Zotkin, and R. Duraiswami, "Virtual autoencoder based recommendation system for individualizing head-related transfer functions," in *WASPAA*, 2013.

[21] Y. Luo, D. N. Zotkin, and R. Duraiswami, "Gaussian process models for HRTF based 3D sound localization," in *ICASSP*, 2014.

[22] Y. Luo, D. N. Zotkin, and R. Duraiswami, "Sparse head-related impulse response for efficient direct convolution," 2014, submitted to IEEE Transactions on Audio, Speech, and Language Processing.

[23] Y. Luo, D. N. Zotkin, and R. Duraiswami, "Gaussian process models for HRTF based sound-source localization and active-learning," 2014, submitted to IEEE Journal of Selected Topics in Signal Processing.

[24] C. Cheng and G. Wakefield, "Introduction to head-related transfer functions (HRTFs): Representations of HRTFs in time, frequency, and space," in *Audio Engineering Society Convention 107*, 1999.

[25] W. G. Gardner and K. D. Martin, "HRTF measurements of a KEMAR dummy-head microphone," *The Journal of the Acoustical Society of America*, vol. 97, p. 3907, 1995.

[26] D. N. Zotkin, R. Duraiswami, E. Grassi, and N. A. Gumerov, "Fast head-related transfer function measurement via reciprocity," *The Journal of the Acoustical Society of America*, vol. 120, p. 2202, 2006.

[27] B. F. Katz, "Boundary element method calculation of individual head-related transfer function. I. rigid model calculation," *The Journal of the Acoustical Society of America*, vol. 110, p. 2440, 2001.

[28] R. Duraiswami, D. N. Zotkin, and N. A. Gumerov, "Interpolation and range extrapolation of HRTFs [head related transfer functions]," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, vol. 4. IEEE, 2004, pp. iv–45.

[29] N. A. Gumerov, A. E. ODonovan, R. Duraiswami, and D. N. Zotkin, "Computation of the head-related transfer function via the fast multipole accelerated boundary element method and its spherical harmonic representation," *The Journal of the Acoustical Society of America*, vol. 127, p. 370, 2010.

[30] D. Zotkin, J. Hwang, R. Duraiswaini, and L. S. Davis, "HRTF personalization using anthropometric measurements," in *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.* Ieee, 2003, pp. 157–160.

[31] Q. Huang and Y. Fang, "Modeling personalized head-related impulse response using support vector regression," *J Shanghai Univ (Engl Ed)*, vol. 13, no. 6, pp. 428–432, 2009.

[32] H. Hu, L. Zhou, H. Ma, and Z. Wu, "HRTF personalization based on artificial neural network in individual virtual auditory space," *Applied Acoustics*, vol. 69, no. 2, pp. 163–172, 2008.

[33] A. Silzle, "Selection and tuning of HRTFs," in *Audio Engineering Society Convention 112*. Audio Engineering Society, 2002.

[34] P. Runkle, A. Yendiki, and G. Wakefield, "Active sensory tuning for immersive spatialized audio," in *Proc. ICAD*, 2000.

[35] A. V. Oppenheim, R. W. Schafer, J. R. Buck *et al.*, *Discrete-time signal processing*. Prentice hall Upper Saddle River, 1999, vol. 5.

[36] D. J. Kistler and F. L. Wightman, "A model of head-related transfer functions based on principal components analysis and minimum-phase reconstruction," *Journal of Acoustical Society of America*, vol. 91, pp. 1637–1647, 1992.

[37] N. A. Gumerov, R. Duraiswami, and D. N. Zotkin, "Fast multipole accelerated boundary elements for numerical computation of the head related transfer function," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 1. IEEE, 2007, pp. I–165.

[38] R. O. Duda, "Modeling head related transfer functions," in *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*. IEEE, 1993, pp. 996–1000.

[39] R. Duraiswami, Z. Li, D. N. Zotkin, E. Grassi, and N. A. Gumerov, "Plane-wave decomposition analysis for spherical microphone arrays," in *Applications of Signal Processing to Audio and Acoustics, 2005. IEEE Workshop on*. IEEE, 2005, pp. 150–153.

[40] N. A. Gumerov and R. Duraiswami, *Fast multipole methods for the Helmholtz equation in three dimensions*. Elsevier Science, 2005.

[41] D. N. Zotkin, R. Duraiswami, and N. A. Gumerov, "Regularized HRTF fitting using spherical harmonics," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2009, pp. 257–260.

[42] J. Kayser and C. E. Tenke, "Principal components analysis of Laplacian waveforms as a generic method for identifying ERP generator patterns: I. Evaluation with auditory oddball tasks." *Clinical Neurophysiology*, vol. 117, pp. 348–368, 2006.

[43] F. Perrin, J. Pernier, O. Bertrand, and J. F. Echallier, "Spherical splines for scalp potential and current density mapping," *Electroencephalography and Clinical Neurophysiology*, vol. 72, pp. 184–7, 1989.

[44] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.

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

[46] C. E. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. Cambridge, Massachusettes: MIT Press, 2006.

[47] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Applied statistics*, pp. 100–108, 1979.

[48] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[49] C. H. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 1, pp. 45–55, 2010.

[50] C. M. Bishop *et al.*, *Pattern recognition and machine learning*. springer New York, 2006, vol. 1.

[51] J. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden Markov models," 1998.

[52] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[53] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," vol. 11, pp. 3371–3408, Dec. 2010.

[54] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[55] B. Settles, "Active learning literature survey," *University of Wisconsin, Madison*, 2010.

[56] M. Osborne, R. Garnett, and S. Roberts, "Gaussian processes for global optimization," in *3rd International Conference on Learning and Intelligent Optimization (LION3)*, 2009, pp. 1–15.

[57] F. Van Loan, "The ubiquitous Kronecker product," *Journal of Computational and Applied Mathematics*, vol. 123, pp. 85–100, 2000.

[58] J. Quinonero-Candela and C. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *Journal of Machine Learning Research*, vol. 6, pp. 1939–1959, 2005.

[59] C. Lawson and R. Hanson, *Solving least squares Problems*. PrenticeHall, 1987.

[60] S. M. inc, *OpenMP API user guide*, 2003.

[61] NVIDIA, *CUDA programming guide 3.2*, 2011.

[62] A. Kulkarni and H. Colburn, "Role of spectral detail in sound-source localization," *Nature*, vol. 396, no. 6713, pp. 747–749, 1998.

[63] E. Wenzel, M. Arruda, D. Kistler, and F. Wightman, "Localization using nonindividualized head-related transfer functions," *JASA*, vol. 94, p. 111, 1993.

[64] G. Romigh, D. Brungart, R. Stern, and B. Simpson, "The role of spatial detail in sound-source localization: Impact on HRTF modeling and personalization." in *Proceedings of Meetings on Acoustics*, vol. 19, 2013.

[65] J. Hornstein, M. Lopes, J. Santos-Victor, and F. Lacerda, "Sound localization for humanoid robots-building audio-motor maps based on the HRTF," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE, 2006, pp. 1170–1176.

[66] M. Rothbucher, D. Kronmüller, M. Durkovic, T. Habigt, and K. Diepold, "HRTF sound localization," 2011.

[67] T. Rodemann, M. Heckmann, F. Joublin, C. Goerick, and B. Scholling, "Real-time sound localization with a binaural head-system using a biologically-inspired cue-triple mapping," in *International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 860–865.

[68] H. Nakashima and T. Mukai, "3D sound source localization system based on learning of binaural hearing," in *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, vol. 4. IEEE, 2005.

[69] V. Willert, J. Eggert, J. Adamy, R. Stahl, and E. Körner, "A probabilistic model for binaural sound localization," *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics*, vol. 36, no. 5, p. 1, 2006.

[70] A. Deleforge and R. Horaud, "2D sound-source localization on the binaural manifold," in *Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on*. IEEE, 2012, pp. 1–6.

[71] F. Keyrouz, K. Diepold, and S. Keyrouz, "High performance 3D sound localization for surveillance applications," in *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*. IEEE, 2007, pp. 563–566.

[72] F. Keyrouz, "Humanoid hearing: A novel three-dimensional approach," in *Robotic and Sensors Environments (ROSE), 2011 IEEE International Symposium on*. IEEE, 2011, pp. 214–219.

[73] F. Keyrouz and K. Diepold, "An enhanced binaural 3D sound localization algorithm," in *Signal Processing and Information Technology, 2006 IEEE International Symposium on*. IEEE, 2006, pp. 662–665.

[74] A. Pourmohammad and S. Ahadi, "TDE-ILD-HRTF-Based 3D entire-space sound source localization using only three microphones and source counting," in *Electrical Engineering and Informatics (ICEEI), 2011 International Conference on*. IEEE, 2011, pp. 1–6.

[75] K. Fink and L. Ray, "Tuning principal component weights to individualize HRTFs," in *ICASSP*, 2012.

[76] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," *arXiv preprint arXiv:0912.3995*, 2009.

[77] C. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Advances in Neural Information Processing Systems*, 2000.

[78] S. Mika, B. Schölkopf, A. J. Smola, K.-R. Müller, M. Scholz, and G. Rätsch, "Kernel PCA and de-noising in feature spaces." in *NIPS*, vol. 11, 1998, pp. 536–542.

[79] M. Seeger, C. Williams, and N. Lawrence, "Fast forward selection to speed up sparse gaussian process regression," in *Artificial Intelligence and Statistics 9*, no. EPFL-CONF-161318, 2003.

[80] R. Saigal, "On the inverse of a matrix with several rank one updates," University of Michigan Ann Arbor, Tech. Rep., 1993.

[81] Y. Saatci, "Scalable inference for structured Gaussian process models," Ph.D. dissertation, University of Cambridge, 2011.

[82] Z. Xu, F. Yan, and Y. Qi, "Infinite Tucker decomposition: Nonparametric Bayesian models for multiway data analysis," in *International Conference on Machine Learning*, 2012.

[83] J. Rougier, "Efficient emulators for multivariate deterministic functions," *Journal of Computational and Graphical Statistics*, vol. 17, pp. 827–843, 2008.

[84] A. Liutkus, R. Badeau, and G. Richard, "Gaussian processes for underdetermined source separation," *IEEE Transactions on Signal Processing*, vol. 59, pp. 3155–3167, 2011.

[85] E. Bonilla, K. Chai, and C. Williams, "Multi-task Gaussian process regression," *Advances in Neural Information Processing Systems*, vol. 20, pp. 153–160, 2008.

[86] O. Stegle, C. Lippert, J. Mooij, N. Lawrence, and K. Borgardt, "Efficient inference in matrix-variate Gaussian models with iid observation noise," in *Advances in Neural Information Processing Systems*, 2011.

[87] L. Baldassarre, L. Rosasco, A. Barla, and A. Verri, "Multi-output learning via spectral filtering," Massachusetts Institute of Technology, Tech. Rep., 2011.

[88] N. Lawrence, "Probabilistic non-linear principal component analysis with Gaussian process latent variable models," *Journal of Machine Learning Research*, vol. 6, pp. 1783–1816, 2005.

[89] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," in *Advances in Neural Information Processing Systems*, 2006.

[90] D. Sorensen and Y. Zhou, "Direct methods for matrix Sylvester and Lyapunov equations," *Journal of Applied Math*, vol. 6, pp. 277–303, 2003.

[91] R. Bartels and G. Stewart, "Solution of the matrix equation AX + XB = C," *Comm. ACM*, vol. 15, pp. 820–826, 1972.

[92] T. Davis and W. Hager, "Row modifications of a sparse Cholesky factorization," *SIAM. J. Matrix Anal. Appl.*, vol. 26, pp. 621–639, 2005.

[93] C. Igel and M. Toussaint, "Rprop using the natural gradient," *Trends and Applications in Constructive Approximation. International Series of Numerical Mathematics*, vol. 151, pp. 259–272, 2005.

[94] G. E. Uhlenbeck and L. S. Ornstein, "On the theory of Brownian motion," *Phys. Rev*, vol. 36, pp. 823–841, 1930.

[95] C. Huang, H. Zhang, and S. Robeson, "On the validity of commonly used covariance and variogram functions on the sphere," *Mathematical Geosciences*, vol. 43, pp. 721–733, 2011.

[96] A. M. Yaglom, "Correlation theory of stationary and related random functions vol. I: Basic results," *Springer Series in Statistics. Springer-Verlag*, 1987.

[97] T. Gneiting, "Correlation functions for atmospheric data analysis," *Quarterly Journal of the Royal Meteorological Society*, vol. 125, pp. 2449–2464, 1999.

[98] W. Zhang, R. A. Kennedy, and T. D. Abhayapala, "Iterative extrapolation algorithm for data reconstruction over sphere," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2008, pp. 3733–3736.

[99] V. C. Raykar, R. Duraiswami, and B. Yegnanarayana, "Extracting the frequencies of the pinna spectral notches in measured head related impulse responses," *Journal of Acoustical Society of America*, vol. 118, pp. 364–374, 2005.

[100] V. R. Algazi, C. Avendano, and R. O. Duda, "Elevation localization and head-related transfer function analysis at low frequencies," *Journal of the Acoustical Society of America*, vol. 109, pp. 1110–1122, 2001.

[101] Z. Botev, J. Grotowski, and D. Kroese, "Kernel density estimation via diffusion," *Annals of Statistics*, vol. 38, pp. 2916–2957, 2010.

[102] S. Hwang and Y. Park, "Time delay estimation from HRTFs and HRIRs," in *International Conference on Motion and Vibration Control*, 2006.

[103] R. O. Duda and W. L. Martens, "Range dependence of the response of a spherical head model," *The Journal of the Acoustical Society of America*, vol. 104, p. 3048, 1998.

[104] J. Quinonero-Candela, "Learning with uncertainty - Gaussian processes and relevance vector machines," Ph.D. dissertation, Technical University of Denmark, 2004.

[105] B. F. G. Katz and D. R. Begault, "Round robin comparison of HRTF measurement system: preliminary results," in *Proceedings of ICA*, 2007.

[106] H. Kuhn and A. Tucker, "Nonlinear programming," in *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, 1951, pp. 481–492.

[107] D. Chen and R. Plemmons, "Nonnegativity constraints in numerical analysis," in *Symp on the Birth of Numerical Analysis*, 2007.

[108] R. Bro and S. Jong, "A fast non-negativity-constrained least squares algorithm," *Journal of Chemometrics*, vol. 11, pp. 393–401, 1997.

[109] M. Benthem and M. Keenan, "Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems," *Journal of Chemometrics*, vol. 18, pp. 441–450, 2004.

[110] D. Kim, S. Sra, and I. Dhillon, "A new projected quasi-Newton approach for the non-negative least squares problem," The University of Texas at Austin, Tech. Rep. TR-06-54, 2006.

[111] V. Franc, V. Hlavc, and M. Navara, "Sequential coordinate-wise algorithm for non-negative least squares problem," Czech Technical University, Tech. Rep. CTU-CMP-2005-06, 2005.

[112] M. Catral, M. Neumann, and R. Plemmons, "On reduced rank nonnegative matrix factorization for symmetric nonnegative matrices," *Lin. Alg. Appl.*, vol. 393, pp. 107–126, 2004.

[113] S. Bellavia, M. Macconi, and M. Morini, "An interior point Newton-like method for nonnegative least squares problems with degenerate solution," *Numerical Linear Algebra with Applications*, vol. 13, pp. 825–846, 2006.

[114] S. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale l1-regularized least squares," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, pp. 606–6017, 2007.

[115] S. Hammarling and C. Lucas, "Updating the qr factorization and the least squares problem," University of Manchester, Tech. Rep. MIMS EPrint, 2008.

[116] A. Bjorck, "Stability analysis of the method of semi-normal equations for least squares problems," *Linear Algebra Appl.*, vol. 88/89, pp. 31–48, 1987.

[117] G. Golub and C. Van Loan, *Matrix Computations*. Baltimore, Maryland: The Johns Hopkins University Press, 1996.

[118] *Math kernel library reference manual*, 2010.

[119] NVIDIA, *OpenCL programming guide for CUDA architectures 3.1*, 2010.

[120] V. Volkov and J. Demmel, "Benchmarking gpus to tune dense linear algebra," in *Super Computing*, 2008.

[121] A. Kerr, D. Campbell, and M. Richards, "Qr decomposition on GPUs," in *GPGPU-2: Proceedings of 2nd Workshop on General Purpose Processing on Graphics Processing Units*, 2009, pp. 71–89.

[122] G. Blelloch, "Prefix sums and their applications," Carnegie Mellon University, Tech. Rep. CMU-CS-90-190, 1990.

[123] M. Harris, *Optimizing parallel reduction in* CUDA, 2007.

[124] A. Bojanczyk, R. Brent, and F. Hoog, "Qr factorization of toeplitz matrices," *Numerische Mathematik*, vol. 49, pp. 81–94, 1986.

[125] R. Chan, J. Nagy, and R. Plemmons, "FFT-based preconditioners for toeplitz-block least squares problems," *SIAM J. Numer. Anal.*, vol. 30, pp. 1740–1768, 1993.

[126] G. Clark, S. Parker, and S. K. Mitra, "A unified approach to time-and frequency-domain realization of fir adaptive digital filters," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 31, no. 5, pp. 1073–1083, 1983.

[127] C. Burrus and T. W. Parks, *DFT/FFT and Convolution Algorithms: theory and Implementation*. John Wiley & Sons, Inc., 1991.

[128] S. W. Smith *et al.*, "The scientist and engineer's guide to digital signal processing," 1997.

[129] J. C. Middlebrooks, "Individual differences in external-ear transfer functions reduced by scaling in frequency," *The Journal of the Acoustical Society of America*, vol. 106, no. 3, pp. 1480–1492, 1999.

[130] D. W. Batteau, "The role of the pinna in human localization," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 168, no. 1011, pp. 158–180, 1967.

[131] V. R. Algazi, R. O. Duda, and P. Satarzadeh, "Physical and filter pinna models based on anthropometry," in *Audio Engineering Society Convention 122*. Audio Engineering Society, 2007.

[132] M. Geronazzo, S. Spagnol, and F. Avanzini, "Estimation and modeling of pinna-related transfer functions," in *Proc. of the 13th Int. Conference on Digital Audio Effects (DAFx-10)*, 2010, pp. 6–10.

[133] D. Seung and L. Lee, "Algorithms for non-negative matrix factorization," *Advances in neural information processing systems*, vol. 13, pp. 556–562, 2001.

[134] C. H. Ding, X. He, and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering." in *SDM*, vol. 5, 2005, pp. 606–610.

[135] N. Gupta, A. Barreto, M. Joshi, and J. C. Agudelo, "HRTF database at FIU DSP lab," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 169–172.

[136] O. Warusfel, "Listen HRTF database," *online, IRCAM and AK, Available: http://recherche.ircam.fr/equipes/salles/listen/index.html*, 2003.

[137] M. Jeub, M. Schafer, and P. Vary, "A binaural room impulse response database for the evaluation of dereverberation algorithms," in *Digital Signal Processing, 2009 16th International Conference on*. IEEE, 2009, pp. 1–5.

[138] E. H. Langendijk and A. W. Bronkhorst, "Fidelity of three-dimensional-sound reproduction using a virtual auditory display," *The Journal of the Acoustical Society of America*, vol. 107, no. 1, pp. 528–537, 2000.

[139] S. G. Johnson and M. Frigo, "A modified split-radix FFT with fewer arithmetic operations," *Signal Processing, IEEE Transactions on*, vol. 55, no. 1, pp. 111–119, 2007.

[140] G. Hinton, "A practical guide to training restricted boltzmann machines," *Momentum*, vol. 9, no. 1, p. 926, 2010.

[141] R. Magnus, Hestenes, and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, pp. 409–436, 1952.

[142] C. Paciorek and M. Schervish, "Nonstationary covariance functions for gaussian process regression," *Advances in neural information processing systems*, vol. 16, pp. 273–280, 2004.

[143] L. R. Rabiner and B.-H. Juang, *Fundamentals of speech recognition*. PTR Prentice Hall Englewood Cliffs, 1993, vol. 14.

[144] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.