A framework for bird songs detection, recognition and localization using acoustic sensor networks

Master’s Thesis

17 February 2006

Author : Vlad M. Trifa

Professors : Alcherio Martinoli and Charles E. Taylor

Supervisors : Chris Cianci
Abstract

An important question in biology that remains unanswered, is the role of animal communication, how and why it emerged, why does it evolve differently even for individuals of the same species, and what factors affect language evolution. To answer this question, this report presents my contribution to an interdisciplinary project whose goal is to shed some lights on the mysterious mechanisms underlying avian communication. Because our focus is to understand all the factors that could have an influence on language evolution in natural habitat, we need non-intrusive tools that allow to record and analyze bird songs so that human influence is minimized. For this purpose, we think that the very active field of sensor networks might provide us ideal platforms to record and process data because of their characteristics that make them suitable for habitat monitoring. In the meanwhile, their distributed aspect allows to correlate environmental factors with bird behavior as no other tools commonly used by biologists allow.

Besides, as we want to understand the role of social behavior in the avian language evolution, I will show that tools derived from human speech recognition can be applied with very good results to recognize bird species and even individuals in real-time. Also, I will present algorithms that can be used to localize birds and track their interactions in very noisy environments, as for example equatorial rain-forests.

We hope that the work presented here will open new perspectives towards novel wildlife monitoring methods that scientists studying animal communication will appreciate for the drastic reduction of time and human effort our tools will allow. However, the design of such tools is a complex task, and requires a wide range of expert knowledge in many different fields.
Résumé

Un des plus grands mystères dans l’étude du comportement animal est le rôle de la communication, comment et pourquoi est-elle apparue, pourquoi évolue-t-elle différemment même pour des individus de la même espèce et quels sont les facteurs qui influencent l’évolution du langage. Pour répondre à cette question, un projet interdisciplinaire impliquant des biologistes, des informaticiens, des électroniciens et des linguistes est en cours dans plusieurs campus de l’université de Californie, aux USA. Ce rapport présente ma contribution dans cet ambitieux projet qui a pour but de créer une palette d’outils destinés à être utilisés par des chercheurs en biologie et en linguistique afin d’étudier la communication chez les animaux, et plus particulièrement les oiseaux, dans leur habitat naturel.

Pour que la présence humaine soit réduite au maximum afin de ne pas biaiser le comportement des oiseaux étudiés, l’utilisation de nouvelles technologies telles que les réseaux de capteurs distribués nous permet d’avoir une plateforme adaptée pour enregistrer et analyser les chants d’oiseaux de façon efficace. De plus, de par leur nature distribuée, ces outils permettent au biologistes une nouvelle approche vers le prélèvement de données concernant le biotope (température, le taux d’humidité, etc) particulièrement intéressante, car ils disposent de ces informations à différents lieux géographiques simultanément. Cette flexibilité leur permet désormais de corrérer ces informations au comportement des animaux depuis un ordinateur portable, sans avoir besoin d’aller sur le terrain, et ceci en temps réel.

De plus, comme l’analyse des chants d’oiseaux est notre centre d’intérêt, nous allons montrer comment des outils qui ont fait leurs preuves dans le domaine de la reconnaissance vocale, ont été utilisés pour ce projet afin de reconnaître les différentes espèces d’oiseaux avec un taux de succès très prometteur. De même, nous allons décrire comment ces réseaux de capteurs peuvent être utilisés afin de localiser les oiseaux, ce qui est un élément essentiel dans l’étude de leur interactions sociales.

Finalement, nous espérons que ce projet pourra ouvrir de nouvelles perspectives vers des méthodes révolutionnaires qui serviront à observer les animaux dans leur biotope d’origine, et que les scientifiques pourront apprécier pour la réduction significative en temps et en effort humain que ces outils pourront apporter.
Acknowledgments

First of all, I’d like to express my deepest respect and admiration to Prof. Charles TAYLOR for allowing me to make my masters thesis under the Californian sun with his motivating and drilling research group. He is an outstanding scientist and an extraordinary man, and it was a real pleasure and enriching experience to work with him. Besides, I’d like to thank very much Prof. Alcherio MARTINOLI for his kind support, and trust he had in me during the time we collaborated together. He has been my enlightening guide on the path to the scientific truth during my master studies. I would also like to thank Prof. Deborah ESTRIN who leads the Center of Embedded Networked System with an iron hand and who gently allowed me to settle at the big black table in the middle of the CENS lab, and Prof. Edward STABLER from the Linguistics department for introducing me the fascinating field of language (even if I could have put much more effort into understanding what he was trying to tell me). Also, thanks to Prof. Kung YAO from electrical engineering, to Prof. Dan BLUMSTEIN who studies marmots, and Prof. Edgar VALLEJO from Monterey Technical Institute, for taking part in this fascinating project.

Then, I’d like to thank to all the formidable people working in Taylor’s lab, which is in alphabetical order: Changde CHENG, Travis COLLIER, Alex KIRSCHEL, Yoosook LEE, Nick MANOUKIS, John MARSHALL, Saul LOZANO-FUENTES, and Yuan YAO. Furthermore, I’d like to thank my official supervisor Chris CIANCI, and Niklaus CORRELL and Jim PUGH, all from the SWIS group at EPFL, for their help and the great time I had while working for Prof. MARTINOLI. Next, I want to thank all the people I collaborated with during this project: Lewis GIROD and Nithya RAMANTHAN for their constant help to get me through the EmStar maze, Hanbiao WANG for his impressive signal processing skills, Martin LUKAC and all the other people from the CENS I might omit, who helped me on this project. Thanks also to Greg KOBELE who loves languages, Ivan GUARDIOLA for his HTK skills, Erika VILCHES and Ivan ESCOBAR both from UTM, Mexico who did an enormous job in extracting features from bird songs, and Andreas MANTIK as well as Chiao-En CHEN from the EE department for their explanations about the acoustic array, and the localization and beam-forming algorithms. Also thanks to Marcos GRIDI-PAPP for writing Soundruler and valuable discussions about detection of bird songs.

A big thanks to Michael HAMILTON from UC Riverside, an outstanding scientist who runs the James Reserve and allowed the CENS to perform live experiments and such an impressive sensor deployment. Thanks as well to Walter KOENIG who is the director of the Hastings Reserve.
Last but not the least, thanks to all the extraordinary people I met in this urbanistic chaos who transformed my presence in the city of angels into a very pleasant and instructive journey. There are way too many of them, so listing them all here would contribute significantly to the deforestation of the amazonian rain forest, and I am not quite sure that my equatorial antbirds would appreciate.

Before I forget, thanks to all the birds on this planet for being happy at any time of day and night and enchant us with their marvelous songs. They contribute to make our lives more joyful (or at least mine) by reminding us that as long as we will hear them singing, there is still hope for a better world where they will be happy to live in!

Above all, I would like to deeply thank my parents for all the sacrifices they made for allowing me to study in Switzerland, then sponsoring my stay at Los Angeles, and especially for their constant support throughout my life. Without their kind devotion for trying to offer me a better life than they had, I'd never have been arrived where I am today.

Two thumbs up for all of you!!!
# Table of contents

1 Introduction .......................... 1
   1.1 Project goals .......................... 2
   1.2 Contribution of this project ................. 3
   1.3 Outline of the report ....................... 4

2 Context of the project .......... 5
   2.1 Overview ................................ 5
      2.1.1 Biology .............................. 6
      2.1.2 Linguistics .......................... 7
      2.1.3 Engineering ......................... 8
   2.2 Bird songs ............................ 8
      2.2.1 Song individuality ................... 11
      2.2.2 Song production and learning ......... 11
   2.3 Our framework .......................... 13

3 Recording and detection ..... 15
   3.1 Data acquisition ...................... 15
      3.1.1 Manual field recording ............... 16
      3.1.2 Internet audio streaming .......... 16
      3.1.3 Acoustic array .................... 16
   3.2 Acoustic sensor networks ............. 17
      3.2.1 The CENS embedded platform .......... 17
      3.2.2 EmStar ............................... 19
      3.2.3 Self-calibration modules .......... 21
   3.3 Signal filtering ...................... 21
   3.4 Bird songs detection ............... 22
   3.5 My implementation ................... 24
   3.6 Discussion ........................... 26

4 Classification ..................... 27
   4.1 Brief introduction to speech recognition 27
      4.1.1 Features ............................... 27
      4.1.2 Features data mining ............... 28
   4.2 Pattern-matching methods ............. 29
      4.2.1 Dynamic time warping (DTW) ......... 29
      4.2.2 Spectrographic cross-correlation (SPCC) 29
Chapter 1

Introduction

"A bird does not sing because it has an answer. It sings because it has a song."

Chinese Proverb

Language is thought to be the key element in the development of the human intellectual properties. Throughout history, scientists and philosophers have always been interested in deciphering the mysteries of communication, and especially understanding how languages originate and evolve, and on a shorter time scale, how a language can be learned by a child. One hypothesis for human language diversity is proposed in the Bible, where it suggested that different languages appeared to prevent humans from constructing the Babel Tower. However, a more realistic hypothesis is that human languages have evolved according to a myriad of factors as geographical location and social interactions. Even though language is an exclusive characteristic of human, many animals are able to communicate, and for them it is a vital process for survival. Many examples shown that animal communication is shaped by evolution, where only individuals that find an optimal communication strategy for a given environment are able to survive. Identification of the factors that influence language evolution could provide significant insights about the neural mechanisms underlying communication, thus making this field of research very interesting for scientists from various domains.

The goal of this project is to create tools that will allow field biologists to study animal communication and behavior in their natural habitats. As it desirable to minimize human presence by using non-intrusive techniques, distributed sensors arrays are a perfect match due to their reduced form factor and the constant evolution of this field. Our design goal is to maximize the robustness and adaptability of these devices in order to reduce human intervention, and operational costs, by enabling remote sensing.
The National Science Foundation (NSF) sponsors an interdisciplinary project\(^1\) taking place mainly at UCLA, that will propose directions towards the creation of a panel of tools that will help scientists to study animal behavior and communication, and to understand how the latter evolves in correlation with factors such as environment and social interactions. This project focuses mainly on birds, as they have been extensively studied and are relatively well known, so they provide an excellent testbed to analyze how acoustic signals are transmitted and perceived in noisy environments, and how the structure of vocalizations could be optimized to achieve these goals. Birds have always been a common source of inspiration used by generations of artists. Literature and poetry have attempted to describe the beauty of bird songs, while many composers directly integrated bird songs in their concertos. Contrarily to common beliefs, birds do not really sing for the sake of pleasure and artistic performance, but rather avian communication is usually the result of an impressive evolutionary process, where natural selection directly operates on the quality of males, where only the most talented sopranos and the healthiest individuals will be selected by females for mating.

1.1 Project goals

Many biologists have been interested in understanding why birds sing, and for this purpose audio data was initially recorded in the field onto video or audio tapes, and that data analyzed in the lab afterwards. With the advent of digital media in the 1990s, more particularly the Digital Audio Tape (DAT) and the Mini-Disc (MD), these new supports have been extensively used for field recordings, because of their compactness, quality and high fidelity.

However, the problem of analyzing the recordings remained, because bird experts had to inspect hours of audio recordings to identify sometimes only some seconds of relevant acoustic events. Furthermore, the demand for these professionals is much higher than the offer, implying extremely high costs to perform this kind of analysis. For that reason, the exponential growth of processing power of computers during the second half of the 20\(^{th}\) century, gave birth to automated speech recognition techniques and made biologists investigate these tools to create software able to recognize bird song automatically. Such systems have been put since the 1960s to prevent bird collisions with planes, and showed different degrees of accuracy, but it is only the recent advances in automatic human speech recognition, especially the use of hidden Markov models (HMMs) that opened new perspectives for efficient application to bird songs identification. Along with the expanding field of sensor networks, we have now the possibility to create revolutionary and powerful analysis tools using low-power embedded systems. Having such a network of distributed sensing devices, allows scientists to observe animals in their natural habitats with non-intrusive techniques, and also these devices can be used in environments where humans have limited access. Finally, automated recognition using such a precise instrumentation, allows researchers to analyze the properties of animal vocalization with much higher precision and sensitivity, in comparison to what human ear can perceive.

The ultimate goal of this project would be to set up a collaborative process where each sensor node sees and understands only a part of the situation, depending on its location, sensing and computational abilities, and to fuse this local and noisy information to form a single and coherent decision making process, where a global and robust knowledge of the situation would emerge from the distributed information sharing process.

\(^1\)See http://www.nsf.gov/awardsearch/showAward.do?AwardNumber=0410438.
1.2 Contribution of this project

In this report, I propose solutions for some of the difficult problems in fully automated bird songs analysis using sensor networks as a step towards realizing the full potential of embedded networked sensing systems. This work reports the first attempt towards building a complete framework for automatic analysis of bird vocalization that integrates recording and detection of bird songs, localization, and recognition of the sound source into a single application using data recorded with distributed microphone arrays.

Meanwhile, automated bird song analysis will provide a standardized terminology for the animal communication research. That will allow experiments to be easily reproduced, and results based on a this ontology could be easily shared and compared among researchers all around the world.

The design of such complex tools is definitely not a straightforward task, since it requires expert knowledge from many different research fields. The work presented here is the fruit of a close collaboration between researchers in biology, computer science, electrical engineering, and linguistics. Also, this project results from the collaboration with others campuses of University of California and Instituto Tecnologico de Monterrey in Mexico. The development of such a framework is a novel tool that biologists will greatly appreciate for its numerous advantages, and in the meanwhile an extraordinary challenge with a practical application for the engineers that design it. I describe what this project could bring to the people involved in this project in the following:

**Biology and linguistics** The tools presented here would be used by field biologists in order to monitor ecosystems using more efficient techniques than the traditional and laborious manual data collection. Biologists will greatly benefit from the distributed aspect of sensor networks, so they can correlate information about the ecosystem collected at different spatial locations, and thus understand much better how interactions between animals and the environment take place. Also, because of their reduced size and their autonomy in terms of communication and energy, sensor nodes allow animal behavior to be analyzed in natural habitat without disturbing the individuals. The data collected will also be used by linguists in order to understand the complexity of acoustic signals and interpolate the cognitive and linguistic ability of the species that generate the signals.

**Engineering** The design of automated bird analysis tools, with all the constraints on size, energy, efficiency, and ease of use, poses very challenging goals that can be attained only with intelligent design of hardware and software. As in the long run we want our system to be easily deployed and to work autonomously in environments where no wiring can be installed and where no power sources are present, we must ensure its robustness against failures that can occur at very different levels. At the same time, it requires the design of energy efficient communication protocols that allow robust information sharing in unpredictable environments, and in the same way collaborative processing algorithms that optimize performance, while minimizing the amount of shared data. Furthermore, it requires development of robust signal processing methods that can extract efficiently signals of interest in noisy environments.
1.3 Outline of the report

This report is organized as follows:

**Chapter 2** details the context where this project takes place, what are the important problems that need to be taken into account for the design of these tools, and what knowledge is required to overcome these difficulties.

**Chapter 3** discusses the different alternatives that are interesting for acoustic recording in the field and what are their respective advantages and drawbacks. Also, the algorithm I used to detect bird vocalization in continuous audio streams is presented.

**Chapter 4** describes the classification and recognition methods I used during this project, and what information can be used to classify bird songs.

**Chapter 5** explains the algorithms used for bird direction of arrival estimation and localization employed for this project.

**Chapter 6** details the different experiments I performed to quantify the performance of each unit composing my framework, namely detection, localization and HMM classification.

**Chapter 7** concludes this project and gives some future directions for improving the results I obtained. Also, I propose how the hardware and software could be optimized to be easily used by biologists.
In this chapter, a brief overview of the global architecture of my framework and of its main building blocks is given. In parallel, I will attempt to give an idea of the complexity of this task, and highlight the difficulties for each component in the system.

2.1 Overview

Our goal is to develop distributed sensing systems that will allow to record and to process bird songs in natural habitats. For this purpose, we need to reach the following goals:

**Step 1:** Develop efficient methods for species and individual recognition and localization using an acoustic sensor network.

**Step 2:** Examine the existence of vocal individuality in birds and determine the features that could allow an optimal classification.

**Step 3:** Correlate the presence of birds and abiotic\(^1\) and biotic\(^2\) conditions.

**Step 4:** Use all this information to understand how the environmental constraints and sociality have influenced the evolution of bird songs.

\(^1\)Abiotic refers to the environmental features such as vegetation, etc.

\(^2\)Biotic refers to the parameters of environment such as temperature, humidity, CO\(_2\) concentration, etc.
Methodologies that can accomplish each of these tasks with a sufficient accuracy in isolation and in simplified environments, such as laboratories, do exist. However, such isolated conditions might fail to highlight the influence of factors that have a capital role in animal behavior in natural environments, as for example presence of predators or limited food resources. Our challenge is to extend the existing methodologies to real, complex and noisy environments and provide a unified framework to perform all these tasks automatically. This goal obliges us to design custom hardware devices that can be easily deployed in the field by biologists who do not have any \textit{a priori} knowledge of the technologies employed, but only of what they are looking for.

Another argument favoring the development of such tools, is that the demand for specialists able to recognize bird species by listening to their vocalizations is high, which implies that processing a large amount of recordings requires a considerable financial effort to gather expert knowledge and to support the time required to analyze the data. Besides, automated analysis has the advantage of creating a "standard and mathematical" definition of bird songs, that would allow a more objective and precise analysis than visual inspection of spectrograms. Finally, it increases drastically the amount of data that can be processed in a certain amount of time.

In order to reach such ambitious goals, collaboration between scientists having very different backgrounds and domains of expertise is required. Some important domains of expertise include:

\subsection{Biology}

Field work performed by researchers in biology and ecology is capital to understand animal behavior, as it is the main source of data. Unfortunately, in many cases, humans have very limited perception, because of the very nature of the environments under consideration. Often, so much is happening at the same time that it is not obvious for the human observer to sort out everything. In our case, knowing what species are present or absent, where, at what time, and what they are doing, and then correlating this information with local environmental properties is essential if one wants to understand the complex interactions that influences animal behavior, and on a larger time scale, evolution.

Knowledge of the avian song system is capital, as that information can be used to design more appropriate tools, that are more efficient as they are adapted to deal with the characteristics of bird songs. Biologists are the experts in birds and they have the experience of field recordings and the particularities of real habitats, such as when to record bird songs, what factors generate noise and what important features are missing in their toolbox.

A collection of distributed sensors could track specific birds in space and time, and thus record individuals movement to study groups migration patterns, locate territory holders, asses species density and elucidate mysteries of intra- and inter-specific interactions. Also, these tools would allow to test hypotheses that are impossible or impractical, concerning birds social systems and the characteristics of avian communication. Such tools could allow them to understand much better the mechanisms of song learning, production, and perception and how birds can distinguish efficiently songs of other birds in extremely noisy environments such as rain forests.
2.1.2 Linguistics

Linguists also need to interpret the collected data to model birds songs using grammar systems. In the long run, we hope that the tools developed for this project will be used to analyze the complex interactions taking place between animals, and to understand the way they influence the evolutionary process of animal communication, as well as the importance of the social system in birds.

One should not underestimate the importance of grasping these principles as they could provide very important results in many domains. From a learning theory point of view, one could understand much better the mechanisms of language acquisition in humans and other animals. This knowledge could explain how a child is able to form a representation of the complex grammatical structure of a human language simply by listening a few number of examples (an extremely tiny subset of the whole language). Children are able to create sentences they have never heard before very early in their developmental process, and this evidence suggest that they form a very accurate internal representation of the grammar of their native language. It is commonly thought ((Sasahara, 2005)), that the grammar itself is not genetically encoded, while the mechanisms used by the human brain to infer the grammar are.

Language generation using grammars, was thought to be a process proper to humans, but recent research (Okanoya, 2004; Sasahara, 2005) shows that finches are actually able to generate their songs according to a regular grammar, which means this process can be modeled by a memory-less finite state automaton (FSM). This result shows that these birds could have a much more complex language than we might think, and their songs are not just the repetition of a fixed sequence of syllables. However, it does not mean that finches create novel phrases to discuss, but only they are capable of increase their attractiveness to females by creating novel songs. We expect our tools will help to answer question such as how this mechanism appeared in the common finch and why. The human language is not a regular language (and thus cannot be generated with a simple FSM), given its capacity to create embedded structures such as recursion, and that makes human language fundamentally different from any other. Another capital result would be to show whether birds are able (or not...) to generate such complex pattern that are closer to human grammar, and if they are, this will imply that birds are capable of recursion. This result will be of the greatest importance in linguistics.

The mechanism which allows language to be acquired in a large population of heterogeneous, distributed and adaptive agents, even in noisy environments where learning can be biased and error-prone, and yet still converge towards a unique common language, is a great question addressed in (Lee et al., 2005b,a). This work provides a simplified framework to perform simulations of language evolution in a population of agents having different degrees of connectivity and network structure.

Science of complexity would also largely benefit from these results by understanding how local interactions between a large number of agents is able to converge towards a stable and coherent communication scheme. These results could be of high importance in the domain of swarm intelligence, as using such mechanism to any heterogeneous groups of distributed agents able to communicate, could lead agents to converge towards a common communication scheme and adapt their communication protocol in an optimal way to the context, so that using symbolic communication the amount of data transmitted could be minimized.
2.1.3 Engineering

In order to provide biologists and linguists all the data they need, we want to design efficient hardware and software tools that perform acoustical monitoring, recording, detection, localization, signal quality improvement by increasing the signal-to-noise ratio (SNR)\(^3\), bird species and individual recognition, and finally parsing the acoustic signal into syllables and modeling of songs using grammar systems. Each of these small tasks, which is a real problem in itself, requires specialists in integrated circuits design to create the hardware and engineers to implement the software to perform it.

Furthermore, as we deal with real, thus unpredictable, acoustic signals, we need specialists that can provide us the signal processing knowledge to devise the appropriate filtering methods. The environments where bird sing, can be very noisy due to parasitic sound sources induced by the reflection and attenuation of acoustic signals by the vegetation, and also by sounds produced by other acoustic sources (water, wind, planes, insects, other birds, etc.). The SNR influences directly the performance that can be achieved by the classification tools we use, implying that we need to find a solution to overcome these difficulties.

Some of the noise can be attenuated simply by using band-pass filters, however a much more efficient method to increase the quality of a signal is the use of beam-forming\(^4\) initially used in military applications. This requires development of application specific microphone arrays, where information collected by multiple sensors is merged in order to reduce the amount of uncertainty about the perceived signals. We will attempt to show that using these techniques are able to efficiently improve the SNR of the signals of interest.

We can already anticipate that this task will face an serious problem of data management and processing. Hours of multi-channel uncompressed audio recordings require to be stored somewhere. In addition, real-time processing of this data requires a large amount of computational power. Besides, as we plan to use embedded networked sensors, efficient communication algorithms need to be developed.

2.2 Bird songs

As birds are central to this project, it is essential to define a standard terminology for bird songs that will be used throughout this report. Birds produce a variety of sounds to communicate with flock members, mates, and neighbors. These sounds vary from short simple calls (also called pulses, syllables, or notes) to surprisingly long songs, which are composed of a complex sequence of syllables. Figure 2.1 shows a bird song of a Barred Antshrike (BAS) to illustrates this terminology. Some important facts and characteristics about bird songs will presented, as knowledge about the particularity and diversity of songs found in nature is essential to understand what difficulties are present for automated analysis systems, and why a unique and optimal solution has not yet been found.

\(^3\)SNR is the ratio between the intensity of a signal and the amount of background noise present at the same time. Thus the higher the SNR, the better, as it means that noise has a low intensity in comparison to the signal.

\(^4\)Beam-forming is a method to create directional filters that focuses their "beam" in a a specific spatial location so that all signals emanating from that particular direction are amplified, while all the signals from other directions are attenuated.
Researchers in bird songs came with the hypothesis that three major factors play a major role in the evolution of bird songs, but very little is known about how these factors contribute to song characteristics:

**Physical environment** Acoustic vocalization is a very efficient communication method for birds, because it does not require visual contact between emitter and receiver so it can be used over long distances\(^5\), in dense forests with high level of noise, and also during the night. On one hand, an acoustic signal must be strong enough in order to reach the other birds and to be understood clearly by the receiver without the need to re-emit the signal and thus extra energy expenses. On the other hand, it must be weak enough to avoid being perceived by predators. This situation suggests that natural selection operates directly upon the optimality of the communication scheme used by the bird, and it is clear that a solution which is optimal in all environments does not exist. It should be pointed that birds use acoustic signals that travel better in particular conditions. For example, in forests where sound bounces off trees and is absorbed by vegetation, brief repeated signals are more appropriate, so if the recipient misses one call, he will surely catch it again.

\(^5\)Songs of some birds such as the kakapo, can be perceived as far as 6 kilometers away.
On the other hand, in open fields birds prefer to sing at high elevation above the ground so as to minimize interference from the ground. This statement is further supported by experimental evidence, where different groups of the same bird species could have a very different language\textsuperscript{6} even if they live in contiguous territories where the environment differs.

**Interactions with other species**  Another factor driving birds language evolution is the interaction with other species, especially between predators and prey. Birds usually do not understand other species vocalization, but it has been seen that other bird species learned to recognize acorn woodpeckers alarm calls, and know that a predator is approaching\textsuperscript{7}. Another example of such evolutionary strategies, is the dominant frequency of an european species of passerines is approximately 8 kHz, which prevents predators to localize them accurately. The radius of the raven and eagle head is too large to use such high frequencies to localize the prey without ambiguity.

**Interactions within species**  A natural question that could arise at this point, is what could birds chat about, and what kind of information is contained in their songs. It should be noted that animal communication has a very important role in many activities that are vital for the survival of the individual. On one hand, short calls are used by both males and females for behavior coordination, as for example foraging, flocking where locating and identifying other mates serves to maintain in-flight association (especially for nocturnal flights), or also predator avoidance where alarm calls are used to notify the other member of the group that a predator is approaching.

On the other hand, the role of long and complex songs is reproduction and territorial defense. A simple example is the mate attraction context, where females prefer to mate only with high-quality males, and the cost of being bluffed by a low-quality male could be high. Females select their breeding males according to the beauty of their song, and more complex and novel songs indicates females that the singing male is more likely to have more developed brain capacities such as memory and navigation, so it is worth mating with that male. Also, songs are used to avoid neighbors to enter a dominant male territory, and a male that has a more complex song, is more likely to gain a territory. This fact suggest a clear evidence of a strong sexual selection mechanism, where the attractiveness of a male is mainly related to its song.

As bird songs are not genetically encoded but learned, more complex songs can be produced via cultural transmission rather than genetical, it should be pointed out that there exists differences of the songs within individuals of the same species. An interesting article (Irwin et al., 2001), shows how gradual divergence in a trait involved in mate choice leads to the formation of new species. They noticed that individuals of the same species of birds located between Siberia and Nepal, form different clusters that have a very different song structure, and are not able to understand each other as shown by playback experiments, thus phenotypic behavior (not genetically encoded) might have caused isolation of sub-populations, leading to increased genetic polymorphism. This difference in their songs may act as an evolutionary drive and that might explain the phenomenon of speciation.

\footnote{Language is a sensitive word that should be handled with extreme care when used in this context. When we refer to bird language, we do not mean language as used by humans, but simply communication protocol.}

\footnote{Source: Yuan Yao, personal communication.}
2.2.1 Song individuality

To allow social interactions, every bird should be able to derive the identity of the individual who produced a song. As pointed out in (Vehrencamp, 2000), in such a context the interest of the sender and receiver are similar, and sender are selected to provide accurately encoded and honest information. Stable and reliable signals can only evolve in conflicting-interest contexts where there is some cost on senders that makes cheating a less optimal strategy than revealing the truth. Cost is a function of the form and design of a signal for the sender, for example the energy expense involved in producing the signal or the risk of attracting predators during singing. Receiver dependent costs can also be distinguished, for example when not paying attention to an alarm call.

For this purpose, the signal needs to encode information about the identity of the emitter in order to allow receivers to choose an optimal action. It is commonly thought that individual vocal recognition is widespread in social birds, and other animal species as well, as for example marmot calls encode such an individual information, as showed (Blumstein and Munos, 2005).

The exact “coding scheme” used by birds is not known by scientists, but several studies shown that many characteristics of bird songs are highly correlated for the same individual, as for example, the length of the silence between two notes, or the duration and dominant frequency of the notes might vary across individuals. It shown in (Sharp and Hatchwell, 2005) that inter-individual variation in songs of long-tailed tits (Aegithalos caudatus) is significantly greater than intra-individual variation. Many other studies also support this hypothesis, called the feature invariant hypothesis, but it should be also noted, that birds do not rely exclusively on these highly individual correlated features, but also use features that are highly variant and do not provide any information to perform efficient classification, as shown in (Nelson, 1989).

In this project we were interested by the Acorn Woodpecker (ACW) (Melanerpes formicivorus) because of its high level of sociality. The ACW is a nonmigratory and group-living bird commonly found in oak woodlands of California. They have an elaborated social systems, in which each individual is engaged in social activities daily. Our hypothesis is that sociality drives the evolution of communicative complexity, and for this purpose each individual should be able to recognize other individuals during their vocal communication. We want to show that variation in the vocalization should exist in the ACW, and use HMMs to show that individuals can be classified efficiently recognized based only on their vocalization.

2.2.2 Song production and learning

Birds have evolved a vocal apparatus, called the syrinx, which is not located in the larynx at the end of the trachea like in mammals, but much closer to the lungs at the bifurcation of the trachea into the primary bronchi. When air coming from the lungs passes through it, the membranes of the syrinx vibrate and generate sounds. The variation in pitch and frequency are controlled by the syringeal muscles which control the rigidity of the membranes, and the more syringeal muscles a bird has, the more complex the songs he can produce. The structure of the syrinx varies across species, and as each bronchus can produce independently a sound, they can produce a far greater variety of sounds than humans can do.

\[^{8}\text{We tested our tools in collaboration with Prof. Blumstein on marmot alarm calls, especially individual recognition using HMMs.}\]
The neural mechanism involved are depicted in (Fee et al., 2004; Okanoya, 2004), where it is hypothesized that vocal sequence generation and learning is as follows: each particular moment in the song motif is associated with a unique population of co-active neurons in the telencephalic sensory-motor integration nucleus, (HVC\(_{(RA)}\)). These neurons drive a brief burst of activity in a small fraction of the neurons in the telencephalic motor nucleus (RA) and in a mesencephalic motor nucleus, the dorsomedialis (DM). The output of the vocal tract is produced by the moto-neurons controlling the activity of the syringeal muscle, through the tracheo-syringeal branch of the hypoglossal nerve NXII\(ts\).

An important question is to know what is learned and what is innate in the song generation mechanism. It is thought that birds learn and refine their songs by listening to adult tutors during a "sensitive" period between 60 to 90 days, as shown by experiment where deafened or isolated birds are unable to produce normal songs. In contrast, birds are still able to produce calls even without the presence of a tutor, which means that calls are hereditary, while songs are learned. After that period, the bird reaches its sexual maturity and its song becomes invariant. Learned songs allows more rapid adaptation of the song to different acoustic environments, and is more efficient in terms of evolution. Learning of vocal sequences is thought to be mediated by the plasticity in the RA (recurrent connections within the RA and input connections to the RA, mainly from the HVC\(_{(RA)}\)), as suggested by the convergence of the anterior forebrain pathway onto the RA.

An explanation that could account for the complexity and individual variety, is the fact that birds learn songs by using auditory feedback, as explained in (Catchpole and Slater, 1995; Kogan and Margoliash, 1998), and this evidence could also explain the emergence of dialects.
2.3 Our framework

The global structure of the framework I developed for this project is presented in Fig 2.3. Each color corresponds to specific step in the procedure, and is explained in detail later in the following chapters. During this project, I focused mainly on implementing the algorithms on a single hardware, and I did not take into account the advantages that distributed and collective systems could bring. The reason is that the algorithms used in this project are not yet sufficiently performant, so it is of higher priority to make them work properly before attempting to optimize their performance by extending them to perform collaborative processing. Nonetheless, I tried to develop these tools with modularity as design goal, so that they are platform-independent, thus can be easily adapted when new hardware will be developed.

I decided to decompose the problem in four independent layers, where each one is activated conditionally by the previous layer. These layers are:

1. **Data acquisition** is the method used to record audio data;

2. **Song detection** is the method used to detect when a bird is singing from a continuous audio stream fed by layer 1;

3. **Source localization and beam-forming** is the method to localize the bird singing a segment detected by layer 2, and then improve the quality of the recorded song using beam-forming.

4. **Song classification** finally attempts to classify the species and/or individual that produced each sound file created by layer 3 and finally store the analysis results.

![Figure 2.3](image-url)
Context of the project
In this chapter I will present the different methods that can be used to record bird songs in the wild. Afterwards, I will describe the algorithm I applied to continuous streams of audio data to discriminate between bird songs and background noise.

3.1 Data acquisition

In this project, four main methods are of interest for recording bird songs:

- **Classic field recordings** on DATs using directional microphones,
- **Live internet streaming** based on omnidirectional microphones (James Reserve),
- **Microphone array** connected to a laptop where data is recorded,
- **Microphone array** connected to a sensor node to process data in real-time.

I wanted to create an interface between the different recording methods and the further stages as transparent as possible, thus the same processing can be applied on acoustic signals, regardless on how they have been recorded. Moreover, it allowed to concentrate my efforts on the design and assembling of the algorithms involved, rather than the development of software optimized for a specific hardware.
3.1.1 Manual field recording

Field biologists usually record animal vocalization using a directional microphone (in our case a Sennheiser ...) plugged into DAT tapes (in our case Sony... sampling frequency 44.1 kHz, sample size 16 bits). However, this method requires the observer to know the location of the animal in order to point the microphone in the appropriate direction. But this requirement can rarely hold when recording into jungle, where vision is limited for the observer due to dense vegetation. For that reason, we are interested in using omnidirectional microphones, so the biologist does not need to point the microphone towards the direction of a singing bird.

Once the songs have been recorded on DATs, they were digitalized and stored on a computer using the software Canari\(^1\). Then, these digitalized wave files have been edited using RAVEN 1.2\(^2\) and MATLAB. Also, we used Soundruler\(^3\) and Sound Analysis Pro\(^4\) in order to measure the different features of the acoustic signal, as explained in the next chapter of this report. The algorithms presented in this report have been developed and tested using a desktop computer running MATLAB using data recorded with this method.

3.1.2 Internet audio streaming

Another method of data collection I mentioned, is to stream the raw recordings in real-time over internet or by satellite, in which case the data would be processed on a remote location using a desktop computer. Such a system has been set up in James Reserve, where an omnidirectional microphone streams continuously audio data. This task could be done as well using a sensor node with a microphone array, but I did not investigate this option. I suggest that the system developed during this project can be very easily adapted to process streamed data in real-time. For this reason, I will not further mention this option in this report.

3.1.3 Acoustic array

The third recording method I mention, is an array composed of 8 microphones arranged in circle that has been developed by our colleagues from the electrical engineering department at UCLA, and throughout this report I will refer to this array as the EE array. The localization and beam-forming algorithms described in Chapter 5 have been initially designed for, and tested using this microphone array. The microphones used are the LinearX M53, which are individually plugged using high quality 3-pin XLR connectors into a PreSonsus Firepod. The Firepod is a complete 24-bit/96k recording studio combining eight high-quality preamplifiers, and is plugged onto a laptop through a firewire connection, where the sounds are recorded using Cubase. Additionally, the microphones need to be powered using a 9V power source, because the preamplifier output on the Firepod has a too large voltage (48V). This complex setup is very impractical for field work, but on the other hand it possesses much more computational power, and the quality of the signals we can get, is proportional to the number of microphones used. We have designed a Simulink model that is able to capture in real-time the 8 channels of audio data synchronously and to store the data in a wave file, but this is used only for short recordings due to poor performance of MATLAB when a large amount of data needs to be processed in real time.

\(^1\)The official Canary website: \texttt{http://www.birds.cornell.edu/brp/CanaryInfo.html}.
\(^2\)The official Raven website: \texttt{http://www.birds.cornell.edu/brp/Raven/Raven.html}.
\(^3\)The official Sound Ruler website: \texttt{http://soundruler.sourceforge.net/}.
\(^4\)The official SAP website: \texttt{http://ofersc.ccny.cuny.edu/html/body_sound_analysis.html}
3.2 Acoustic sensor networks

Nevertheless, even if all these recording methods have their importance for our project, especially to develop and debug the algorithms, one of them is particularly more interesting than the other ones. As suggested by its title, this project focuses on the use of acoustic sensor networks, because having a distributed data collection system endowed with relatively powerful computational properties is very appealing for non-intrusive monitoring methods, as sensors can share their perception (and knowledge/experience) to have a robust idea of what is happening at different locations in the environment. This is why we used a very interesting platform designed especially for this application, which I will describe now.

3.2.1 The CENS embedded platform

The desirability of distributed sensor arrays is obvious, as collaboration among nodes provides a much more robust system that is able to face the unexpected situations that might arise in real environments. Also, such a distributed system enables better localization, provides an increased fault tolerance, and facilitates data sharing to optimize resources usage.

In order to reach these goals we need a platform that is able to process efficiently a high amount of data in real-time. The Center for Embedded Networked Sensing (CENS) at UCLA has developed such an embedded platform based on a low-power main-board named the Stargate\textsuperscript{5}. Development of software to perform autonomous bird songs classification and localization has been one of the main target of this project, and my role has been to create this software while keeping in mind what is the data needed by the biologists and linguists.

These sensor nodes can be easily and rapidly deployed in real environments, as all the hardware is self-contained in waterproof cases (see Figure 3.1). Each case contains a Slauson motherboard based upon the Intel PXA255 400 MHz processor, with 64 MB of RAM and 32 MB of onboard flash and has 2 PCMCIA ports. Besides, there is also an SD-card slot on it where we plugged 2 GB cards. Sound acquisition is done using a VX Pocket 440 PCMCIA sound board, which has 4 input channels. Wireless communication is done using and a 802.11B WiFi (SMC2320W) card with an external antenna. The advantage of using a high-power radio source (as 802.11 in this case) should not always be disregarded in comparison to a low-power radio emitter (Zigbee), as we must take into account the different sources of energy consumption, as for example the initialization of the device. Also, higher power radio also means a higher bandwidth, so their energy consumption per unit of information transmitted (J/bit) is preferred, especially when we want to transfer large amount of data. Unfortunately, the autonomy of devices when operated using a 7500 mAh Panasonic battery, is only approximately 24 hours, which makes them inappropriate for long-lived deployments.

At this point, it is important to contrast these sensor nodes with the well-known Crossbow Mica motes, where optimized energy consumption and a reduced form factor are the design goals, while processing power is secondary. In contrast, our platform concentrates on the abilities of individual nodes, due to the nature of data to deal with (high frequency multi-channel audio processing). For a more detailed discussion about these aspects, see (Trifa, 2004).

\textsuperscript{5}Actually, the first version was based on the Stargate motherboard made by Crossbow Corporation, but on the ones we worked on were based on the Slauson, made by Sensoria Corporation. Both these terms will be used indifferently to refer to the new version based on the Slauson.
The audio recording is done with an external board containing the microphone array (see Fig. 3.2). This array is composed of 4 hearing-aid condensers (RTI1207A) having a flat response between 50 Hz and 15 kHz, which are pre-amplified by a custom made integrated circuit. Besides, the array contains four piezo tweeters that are used for the self-localization process. Unfortunately, these loud-speakers have a bad frequency response under 2 kHz and their low quality amplifier is very noisy, making them not optimal for playback experiments.\(^6\)

As a final goal, we would like our system to be:

- **collaborative**, through communication to reach a collective understanding of the environment and events occurring there by collecting data at different locations;
- **robust**, in that they can handle changing environments or agents and the global system remains unaffected by occasionally wrong or noisy messages;
- **adaptive**, in that they can learn to deal with unexpected events, form new concepts and communicate with protocols that are specialized for particular agents;
- **self-configuring**, to be deployed in unknown environments with unknown properties, and deal with changing situations and goals.

Obviously, communication among peers in sensor networks is unreliable, and constrained by low bandwidth, and limited processing capabilities of the nodes. Thus, we need to design methods that reach efficient communication and an adaptive communication scheme would be a highly efficient solution, as suggested in (Vallejo and Taylor, 2006).

\(^6\)We refer to experiments where recorded bird songs are played through the speakers, in order to analyze the responses it generates in the other birds. However, we could easily replace them by high quality speakers as our initial choice was financially motivated.
3.2. Acoustic sensor networks

Figure 3.2: The microphone array used with the Stargate embedded platform. Left: The microphones are horizontally separated by a distance of 8 cm, and the 4th mic is elevated of 14 cm relative to the others in order to perform 3D localization.

3.2.2 EmStar

To make this hardware function under the severe constraints imposed by nature, we need an efficient operating system that is optimized to run on embedded sensor nodes, and that also has integrated primitives to facilitate the development of distributed applications. For this purpose, a specific software environment called EmStar, has been developed at the CENS and is suited to be easily integrated onto any embedded hardware running under Linux.

EmStar is a comprehensive software framework especially designed for the development and deployment of heterogeneous distributed sensing applications. It provides a whole range of tools for simulation, visualization, and debugging of real and simulated sensor networks. Along with this, it provides many services, such as transparent networking and precise time synchronization across multiple nodes. EmStar and the Stargate platform are described in detail in (Girod et al., 2004; Girod, 2005a,b; Wang et al., 2005b). Nonetheless, we will briefly describe some of the interesting properties that makes EmStar perfectly suited for our project.

IPC-FUSD

An EmStar based system is composed of a myriad of different modules interacting together using Linux inter-process communication (IPC) primitives. IPC is done using the Framework for User-Space Devices (FUSD)\(^7\), which is a micro-kernel extension for Linux that allows device-file callbacks to be proxied into the user-space that can be accessed from user-space programs instead of kernel code. This makes the design of a distributed system very modular, and is

\(^7\)For more information about FUSD, see the official website: \texttt{http://www.fusd.com}
in a way similar to Tiny-OS where different components are linked together. Unlike Tiny-OS, which is optimized to run under very limited amount of computational resources, EmStar is suited for much more powerful hardware with higher data storage, allowing us to run more complex algorithms and image/audio processing. While Tiny-OS does not allow multiple process running simultaneously, EmStar benefits from an increased robustness due to fault isolation via multiple processes, so that processes can be created, destroyed and reconnected together without affecting the global stability of the system. Also, IPC allows the system to be more reactive, as event-driven software structure with asynchronous system calls can be used.

EmStar implements a design pattern called StatusDevice, that reports the current status and configuration of a module both in human-readable and binary form. This information is stored into a file somewhere in the file system (usually /dev/) and can be accessed by using the standard linux command cat. This is very useful and has multiple purposes, as for example to be able to access the status of the neighbors table or the radio link statistics. This method greatly simplifies the debugging of distributed applications by adding a high visibility into the system, where all services (modules) can be accessed interactively from the command line terminal. The obvious advantage is that it makes our system more robust and reactive to dynamics of the environment, as all services can be reconfigured at runtime.

**Synchronous sampling**

Another very useful design pattern included in EmStar, called SensorDevice, provides a very convenient interface to access data from other processes and hardware devices. A so-called server acquires the data from the hardware and calls a function to push it on the interface where it is stored in a ring buffer that keeps track of the recent data by assigning a monotonic index to each sample. This data can be further accessed by multiple clients by requesting a range of samples to the sensor device, using a query or a callback function. A query would be "Give me 10000 samples recorded 4 seconds ago.", while a callback function would be "Call the function X each time 1000 new samples arrive in the buffer".

The audio sampling is implemented using SensorDevice, where the 8 last seconds of the audio data recorded by the 4 microphones are stored into such a server. It should be pointed out that the synchronization discrepancy between the 4 channels on the CENS is in the order of 2-5 µs, while on the Firepod channels for the array described in Section 3.1.3 is 65 µs.

**Time synchronization**

The ability to relate the times of events on different nodes is critical to most distributed sensing applications, especially in our case as we need to correlate high-frequency phenomena. EmStar integrates a module called TimeSync that implements a mechanism to interpret the time-stamp (or CPU clock time) from one node on another node using dynamic conversion parameters. The conversion factors are estimated using linear regression to correlate the arrival times of broadcast packets at different nodes. This is a very interesting mechanism in comparison to a single master clock, as it preserves the scalability of a system. Also, precise synchronization between nodes is vital to correlate timing of events for collective processing.
3.2.3 Self-calibration modules

In order for the system to be easily used by biologists, we want to build rapidly self-deployable and self-configuring sensor nodes, and as acoustic source localization is performed, we would like it to be also self-calibrating (determine the relative orientation of the sensor arrays). One way to determine node location would be to use Global Positioning System (GPS), but this method has many drawbacks, as foliage in the jungle or reverberation in canyons reduces drastically the signal from the satellite. Also, this equipment is expensive, consumes a large amount of energy, and the initialization of the system takes time. Finally, these systems are huge and heavy if we want to have an accuracy within a centimeter, and the US government does not allow the user to access precise positioning data, but only 24 hours after it has been collected.

Because of the difficulty to obtain and maintain manually the location and orientation of the arrays in the field, EmStar integrates a module that performs self-localization and self-calibration in three dimensions of the microphone arrays using the time of flight (TOF) of acoustic signals. This process can be run periodically to re-calibrate the arrays in case if the system components are bumped or moved, besides it allows an impressively precise localization of nodes, and is discussed in detail in Lewis Girod’s Ph.D. thesis (Girod, 2005b). This work presents in detail the EmStar framework and shows that accurate localization can be performed with a prototype version of the microphone array. This might be encouraging to expect similar localization performance when localizing birds. However, this system has a priori knowledge of the exact form of the signal used to determine distance, which is not the case with bird songs.

3.3 Signal filtering

Field recordings can be very noisy, and the noise level can become very severe, as for example in tropical rain forest due to environmental characteristics such as wind, rain, sound reverberation due to vegetation, or also due the cacophony produced by other animals present in the environment.

In the case where we want to detect a single bird vocalization that has a narrow band spectrum, it is very easy to design a band-pass filter centered around that frequency so all other frequencies are filtered out. However, in our case, as we want to deal with multiple birds each one having a potentially different spectrum, the problem becomes much more complex. One solution would be to create a single band-pass filter that covers all the frequencies contained in the vocalizations of the birds of interest. Birds songs use rarely frequencies under 1 kHz, whereas most of noise due to wind and microphone handling is usually low frequency, thus high-pass filtering frequencies above 500 Hz is appropriate. On the other hand, antbird songs have rarely a dominant frequency above 10 kHz and in the same way a low-pass filter can be used, resulting in a band-pass filter with a wide band.

The other solution would be to create a narrow band-pass filter for each of the species, apply them sequentially on the recorded signal, and perform detection on each of the resulting filtered signals. Obviously, this is not a feasible solution for an embedded system, as with an increasing number of species under consideration the amount of computation and storage needed would become quickly too large to be handled by the restricted computational power of embedded systems.
3.4 Bird songs detection

The problem of telling if a specific bird is singing at a specific time in such noisy environments, has delved scientists for many years, and no solution that works perfectly for any noise levels has yet been proposed. Such a task is very complex, and a great part of this project has been devoted to finding a solution to this problem.

Constant false alarm rate detection (CFAR)

The algorithm I implemented has been used in many applications, and the pseudo-code is given in Algorithm 1. Basically, it allows to identify high energy segments in continuous streams of audio data, and I used it in order to segregate between bird songs and background noise. The algorithm first estimates the statistical distribution of the amount of energy in a specific frequency range contained in the ambient noise on \( N_t \) consecutive samples (we assume that noise follows a normal distribution \( N(\mu, \sigma^2) \)). Afterwards, the energy present in the same bands in the incoming data is monitored, and a threshold function detects when the power changes significantly from a statistical point of view, that is when the energy of the current segment exceeds the threshold defined \( \mu + \beta \cdot \sigma \), where \( \beta \) is a parameter. In my experiments I used generally \( \beta = 3 \), but when the amount of background noise is high, as in jungle recordings, I had to use \( \beta = 2 \) or even \( \beta = 1.5 \), which works fine only if the noise level is constant.

The way the distribution of the noise power spectrum is estimated, is a critical factor for efficient detection. If the noise distribution is initialized with data containing bird songs, the variability will be too high as it will think that the bird songs are part of the noise, and the threshold will be too high to be reached, resulting in no detection. On the other hand, if the noise used to initialize is too low according to the real noise level present, every single faint sound, will contain energy above the threshold and will be detected as a song. To make things even more complex, noise in real environments usually varies significantly over time (rain starts and stops, other animals making sounds, etc), in which case it becomes appealing to update the noise distribution as it varies. For this purpose, expectation windowed moving average (EWMA) can be used to update mean and variance of the noise power as follows:

\[
\begin{align*}
\mu_{t+1} &= \alpha \mu_{\text{new}} + (1 - \alpha) \mu_t \\
\sigma_{t+1} &= \alpha \sigma_{\text{new}} + (1 - \alpha) \sigma_t
\end{align*}
\]

Where \( \alpha \in [0, 1] \) is the changing rate, where \( \alpha = 0 \) means no update at all, and \( \alpha = 1 \) changes the distribution immediately, without any memory of the previous noise level. One must be extremely careful when using this method, to not change the noise when birds are singing. Also, a low value for \( \alpha \) should be used, as we want to avoid to consider spurious and short sounds as part of the background noise and use this insignificant events to update the noise distribution.

It must be pointed out that this detection scheme could be said to be naive, as it uses only the presence of high energy in a specific frequency range, and detects when the energy varies significantly from its level at rest. This implies that any acoustic signal having sufficient energy in that frequency range will be considered as a bird song, and during my experiments I noticed that some segments of human speech have been misclassified as bird songs. Using such a simple and "naive" algorithm has a very limited use in terms of inference of sound source’s nature, as it does not take into account any information about the specific temporal and spectral structure of bird songs.
Algorithm 1 CFAR Algorithm adapted for bird song detection

Require: $\mu =$ average of ambient noise energy, $\sigma =$ standard deviation of ambient noise energy

1: $state \leftarrow 1$
2: loop
3: /* fetch the next $N$ samples from the audio server ring buffer */
4: $DATA \leftarrow \text{POP(buffer)}$
5: $t_{\text{now}} \leftarrow \text{TIMEINDEX(DATA)}$
6: /* PSD is the power spectral density function */
7: $E \leftarrow \text{PSD}(DATA)$
8: if $E \geq \mu + \beta \cdot \sigma$ then
9: /* This is a high energy segment */
10: if $state = 1$ then
11: /* Start recording */
12: $state \leftarrow 2$
13: $t_{\text{beg}} \leftarrow t_{\text{now}}$
14: else if $state = 3$ then
15: /* Vocalization restarted, keep recording */
16: $state \leftarrow 2$
17: end if
18: else
19: /* This is a low energy segment */
20: if $state = 2$ and $t_{\text{now}} - t_{\text{beg}} \leq l_{\text{note}}$ then
21: /* Pulse is too short so it is a spurious interference. Discard it. */
22: $state \leftarrow 1$
23: else if $state = 2$ and $t_{\text{now}} - t_{\text{beg}} \geq l_{\text{note}}$ then
24: /* Maybe the end of vocalization, but not sure so wait $l_{\text{sil}}$ to see if song is over */
25: $t_{\text{end}} \leftarrow t_{\text{now}}$
26: $state \leftarrow 3$
27: else if $state = 3$ and $t_{\text{now}} - t_{\text{end}} \geq l_{\text{sil}}$ then
28: /* Silence too long, end of the vocalization */
29: if $t_{\text{end}} - t_{\text{beg}} \geq l_{\text{song}}$ and $E \geq E_{\text{min}}$ then
30: /* The segment is long enough, it is surely a song, so store it */
31: $state \leftarrow 1$
32: end if
33: end if
34: end if
35: end if
36: end if
37: end loop

To avoid false positives, one needs to integrate other simple features that could be used as discriminant function. The problem of deriving the nature of acoustic signal is called auditory scene analysis (ASA), and many research has been done in that domain. Several different methods to classify sound signals with very different degrees of accuracy and computational complexity have been proposed. Some of them are described in (Clarkson et al., 1998), where it is pointed out that from the large amount of features that could be extracted from an acoustic signal, only a few of them are sufficient for correct classification.
As we plan to run our algorithm on the embedded platform while limiting the energy expenses for monitoring, my choice was to keep the detection algorithm as simple as possible to perform a coarse grained classification. The only requirement I had, is that I want to keep rejection of true negatives (discard bird vocalization by classifying them as noise) as low as possible, even if the price to pay is a larger amount of false positive acceptation (record noise which is classified as a bird song). The fine-grained classification will be performed by hidden Markov models. However, this is not an optimal choice, because if noise is interpreted as a bird song, the HMMs will not tell that it is just noise and not a bird song, therefore the noise will be recognized as the bird model that has the highest probability to generate the noise (even if the probability is very low, see next chapter for more explanations about HMMs).

The performance of this simple detection algorithm can be easily improved by taking into account some basic information about the structure of a bird song. This is what I did, as I defined a song as a set of short high-energy pulses having a minimal duration of \( l_{\text{note}} \) ms, separated by a silence of maximal length \( l_{\text{sil}} \) ms. The length of the set of pulses (which is the whole song) should have a minimal length of \( l_{\text{song}} \), and a minimal amount of energy \( E_{\text{min}} \) should be contained in the vocalization in order to classify it as a song. This will allow to discard incomplete songs, and at the same time vocalizations that are too weak because the individual is too far away. As soon as an acoustic event that fits these constraints is detected, the whole segment is stored into a sound file that will be fed into the HMMs. Much more complex detection algorithms can be used, using more of the information contained in the signal, as for example cross-correlation of spectrograms or the use of discriminant functions. The detection performance would be higher, but also the processing power required. For this reason, I decided to use this detection algorithm as it can be easily implemented on sensor nodes with limited processing abilities.

Another important technical aspect that is worth mentioned, is as that the spectral energy is being monitored in real-time, a Fourier Transform also needs to be performed in real-time using very limited computational power. Unfortunately, even an efficient Fast Fourier Transform (FFT) as proposed in Numerical Recipes in C (Press et al., 1992) is not efficient enough, due to absence floating point co-processor on the Slauson motherboard. This forced us to use an alternative implementation that uses fixed-point computations. The reduction in time required for processing by this alternative method is highly significant, but the price to pay is a reduced accuracy, and discrepancies with the results of the MATLAB implementation of the FFT are high, which makes difficult the comparison of the same algorithm in the MATLAB framework and the implementation on the Stargate platform, which would facilitate debugging of the code.

### 3.5 My implementation

At first, I wrote a simple software that stores continuous recordings of the data coming from the sound card in binary format. After testing this first version, I noticed that the SD card where the data had to be stored, is too slow to handle writing of such a large quantity of data as it arrives. It allowed to write approximately 2 minutes of data before the write buffer is filled and the application crashes. In order to handle writing of 4 data channels sampled at 48 kHz in real-time, this version has been modified, and a thread that compresses that data
Figure 3.3: The workflow of the program I developed during this project. The Audio Server task is to store all the incoming audio data (the data source is chosen when the server is started) into an indexed ring buffer. Then every time a new block of data arrives from the hardware, it triggers the FSM implementing the CFAR algorithm that will monitor the block of data to see whether a vocalization is contained or not.

Every time a segment containing a bird song is detected, the data is read from the ring buffer in the audio server by the localization module. Each detected segment is decomposed into different sources, and for each of them, we create a filtered sound file that will be processed by the HMMs in order to identify the source.

using FLAC\textsuperscript{8} encoding has been integrated, in order to reduce the amount of data to write on the memory card. However, the improvement when using this modified version was not really significant as the data needs to be compressed prior writing. As the Linux kernel version used by the embedded platform was not the most recent one (currently installed kernel version is 2.6.10, and the latest kernel version available for the embedded platform is 2.6.14, but has not been installed), and given that the SD card was the most recent model available on the market, we suggest that upgrading the kernel version will fix this bottleneck by allowing us to use the maximal speed the hardware can handle. Besides, I tried to write the data directly on a remote computer using the NFS protocol, both through wireless and ethernet interfaces, but no improvement in the amount of data that can be written has been noticed. The latest version written allowed to store over 10 minutes of sound when recording using a single microphone, but no more than 3 minutes could be recorded when all 4 audio channels were stored. We noticed that data compression did not significantly affect the amount of data we were able to write before the application crashes. For that reason, I did not further attempt to implement the whole framework on the CENS platform, but I only used it to record the data that was processed by the framework in MATLAB.

Besides, I wrote a simulator that allows to test the software offline, where the data recorded with the microphones is read from a file instead of the hardware. My software creates another device file (implementing SensorDevice) that has the same interface as the device file associated with the microphones. This allows to test the algorithm using the same file on a desktop

\textsuperscript{8}FLAC stands for Free Lossless Audio Compression, that allows data compression without loss of information contained in the signal. See official website \url{http://flac.sourceforge.net}.
computer and on the embedded platform. Also, a version of the CFAR algorithm has been implemented on the CENS node, but has not been tested with real data.

However, to test my algorithm on a desktop computer, as I did not attempt real-time detection because the CFAR algorithm does not detect perfectly, I decided to record everything in the field without any data processing, and then process the recorded files offline on a desktop computer to ensure that no bird vocalization might be discarded by the detection algorithm. I did not use the concept of an audio server in the MATLAB code I wrote, and the data is read from a wave file.

3.6 Discussion

One great advantage of EmStar, is that the audio server is directly available when using the SensorDevice pattern. Also, the triggering of other modules can be implemented using the StatusDevice, for example when the detection algorithm finds a song starting at $t_1$ and finishes at $t_2$, it will post a new event in its StatusDevice file. This will trigger the localization algorithm that will localize the sound starting at $t_1$, and it will post the localization results in its own StatusDevice file. This second event will trigger the beam-forming algorithm that will process the detected vocalization using the estimated angle that was posted by in the localization StatusDevice, and it will output the resulting sound file, that will trigger in the same way the HMM recognition to process the resulting file.

We suggest that EmStar is a very promising software framework that is worth being investigating for development of a new generation of motes, that are able to meet the requirements of collaborative processing systems. A lot of effort has been put into the design of this platform, and similarly to Tiny-OS for Mica nodes, EmStar could become the standard de facto for more powerful sensor nodes.
Chapter 4

Classification

"Fall is my favorite season in Los Angeles, watching the birds change color and fall from the trees.”

David Letterman (1947 - )

Nowadays, speech recognition is a reliable and widespread technology, and has been successfully employed in many industrial applications. This chapter describes different methods and tools from this area, and also how I used them in order to recognize bird songs.

4.1 Brief introduction to speech recognition

There are two main categories of speech recognition architectures. On one hand, the knowledge-based approach where one attempts to encode the knowledge that an expert phonetician will use to decode a signal, where different features such as phonemes or words are extracted from an acoustic signal. On the other hand, the pattern-matching approach where information about the rules of the language or phonemes is required. In contrast to knowledge-based systems, these systems acquire their knowledge through training on a large databases of examples, or by unsupervised learning methods. The following sections present briefly HMMs as an example of knowledge-based system, and a brief discussion about the main differences with pattern-matching procedures, such as dynamic time warping (DTW) or spectrographic cross-correlation (SPCC), is given.

4.1.1 Features

Acoustic signals contain many different attributes, called cues or features, that can be correlated or not with the production of a specific sounds and/or across species. The amount of
features that can be extracted from a signal at a specific time is virtually infinite, ranging from the frequency of the signal to the logarithm of the third time derivative. However, it is less likely that birds rely on the latter to encode information. As discussed previously, birds encode information in their songs, but not much is known about how they do it, so we will subdivide the features we think they use in two main categories: temporal and spectral. Temporal features are for example duration of a note, length of the silence between notes, length of the song, number of notes per song, etc... and it has been shown that might they differ significantly across individuals.

Unfortunately, using only temporal features is not sufficient to discriminate efficiently, as other features are also used to encode the information. Spectral features are related to the different frequencies contained in a signal, as for example the amount of energy in a specific frequency band. For information only, I included in Appendix B a set of features commonly used in bioacoustics that can be measured using the software Soundruler. However, this is only a small subset of all features that might be used by animals. Other higher-level features could be related to the syntax and ordering of the syllables. Using such features can be useful to improve the recognition performance. As an example, in speech recognition, classification of a syllable can be improved when looking at the amount of energy in a speech segment, if there’s a high amount of energy it is very likely to be a vowel, otherwise it is a consonant.

Unfortunately, most features can change during conflict situation when other birds intrude a bird’s territory. Singing rate (songs per minute), duration, amplitude, frequency, and complexity vary during aggressive encounters, and this change might be perceived by the other members of the group who will interpret it as an alert. These features also differ according to the size and health of the bird. This variation in individual songs makes difficult to perform classification by using absolute values for these features.

4.1.2 Features data mining

Given such a large set of features that can be used for classification, the obvious question is which of them should be used in order to have a maximal separation between individuals and/or species? An interesting article (Nelson, 1989), shows that birds do not rely on the complete set of invariant features to recognize other birds of the same species, and on the other hand they also rely on other features that vary. This might be the case when birds recognize their family members, but in our case we need to have concrete methods that allow the best discrimination given the data at our disposal.

The work done by our collaborators at the Instituto Tecnologico de Monterrey (Mexico) and described in (Escobar et al., 2005; Vilches et al., 2006) is a very useful contribution to this project, and bird study in general. They have measured manually approximately 70 features contained in thousands of recordings using Soundruler, and then they used data mining and the ID3 algorithm to create a hierarchical tree of the features that allow classification with the highest degree of confidence. In other words, they look what features allows to clusterize optimally individuals (the features that reduce the most the entropy), similarly to what does principal component analysis (PCA), and they order the features according to their discriminative power. The advantage of this process is that the amount of features to look at can be drastically reduced (they reduced the 70 features to 15) with almost no loss in recognition performance.
Using data mining techniques is highly interesting, as it can provide unique information on which features are the most correlated for the same individual, and at the same time most differ across individuals, and this allows an optimal clustering of individuals, as performance depends on the degree of variability within and separability between classes in the feature space. Data mining requires a lot of data, which is a long and painful process, so it could be interesting to automatize the data collection process. It should be noted that Soundruler has such an automatic mode to compute these features automatically, but it rarely detects correctly every syllable in a bird song, as it is mainly used to analyze highly stereotyped animal vocalizations such as frogs calls, and is not suited to be used for bird songs. For this reason, my detection algorithm can be interesting for detecting songs in continuous recordings and automatically extract the features of interest. However, my algorithm has not been tested to detect syllables, but only complete songs.

The weakest point concerning the tree generated by the algorithm ID3, is that when a new species is added to the database, the whole tree might be reorganized as other features might allow optimal (in the sense of maximal entropy reduction) classification when the new bird is taken into account, and that can lead to a complete redesign of the whole classification procedure. But on the other hand, the information given by data mining is highly valuable to understand the correlation between acoustic cues and individuals/species that would allow maximal classification efficiency.

### 4.2 Pattern-matching methods

Many existing algorithms and statistical tools can be applied to solve the problem of finding the most similar song into a set of samples, and any given unknown class. The performance of these procedures can vary according to their computational complexity. Unfortunately, these methods rarely deal with the dynamic aspect of bird songs, and small variations between the spectra of the two songs can have a large impact upon performance. Two methods that have been applied for bird songs recognition are:

#### 4.2.1 Dynamic time warping (DTW)

In contrast to statistical systems as HMMs where the constituent vocalizations to be recognized are estimated through training, DTW systems labels an acoustic input using a pre-specified set of template patterns of bird song constituents. The matching is then measured according to the Euclidian distance between the magnitudes of the Fourier transform bins of the vectors of the input signal and of the template. This method, described in (Kogan and Margoliash, 1998) minimizes the total distance between the sequence of warped templates and the continuous input, and constitutes an efficient mechanism to compensate for the variability in length of sounds.

#### 4.2.2 Spectrographic cross-correlation (SPCC)

This method has been used in (Sharp and Hatchwell, 2005), and it simply consists to perform a cross-correlation between the spectrograms of a reference signal and an unknown signal to identify. However, if the length of the spectrograms to compare is slightly different, incorrect classification might occur.
4.3 Features extraction methods

The key idea of the knowledge-based approach requires to analyze the signal in order to extract the value of a set of features at regularly spaced discrete time step, and use this information to compare songs. The key idea is to represent a signal using a sort of parametric form, and to compress the amount of data needed to store the whole signal by extracting only the relevant information out of it. For example, a vector of features (let’s say the dominant frequency and the energy) is extracted from a recorded signal every 10 ms (each of these vectors will be called an observation), and this will result in a time series. Afterwards, an HMM for each class to recognize (in our case a HMM per bird species or individual) is used to model the temporal evolution of the features of its class, and recognition is done by looking at which HMM is the more likely to produce the sequence of observations. The ability to model such time series is what makes HMMS interesting as opposed to pattern-matching methods, as length of the songs can vary and still be recognized correctly.

**Hamming window** Usually, it can be useful to apply to the samples a windowing method in order to decrease the incontinuities of the window as stated in (Young et al., 2002). This is done using a Hamming window, and the following transformation is applied to the samples:

\[ x'_n = \left( 0.54 - 0.46 \cos \left( \frac{2\pi (n - 1)}{N - 1} \right) \right) x_n \]  
(4.1)

4.3.1 Linear prediction coefficients (LPC)

As usually, there exist a correlation between the samples in a segment of acoustic data, the idea use by LPC is to encode such a segment as a set of coefficients in a given equation that allows to predict the value of all samples according to the previous ones. The coefficients are chosen so that they minimize the error error between the real values of the samples and the predicted values when using the coefficients and the equation. A more theoretical description is given in (Young et al., 2002). This method attempts to model the vocal tract transfer function as using an all-pole filter with transfer function:

\[ H(z) = \frac{1}{\sum_{i=0}^{p} a_i z^{-i}} \]  
(4.2)

where \( p \) is the number of poles and \( a_0 = 1 \). The filter coefficients are chosen to minimize the mean square prediction error summed over over the analysis window by using an autocorrelation method as follows: given a window of samples \( x_n, n = 1, N \), the first \( p + 1 \) terms are computed as follows.

\[ r_i = \sum_{j=1}^{N-i} x_j x_{j+i} \]  
(4.3)

where \( i = 0, p \). Then we use a set of auxiliary coefficients \( k_i \) in order to compute the filter coefficients recursively over \( i \) as follows: first derive a new coefficient of \( i^{th} \) order \( k^{(i)} \) using \( k^{(i-1)} \):

\[ k_j^{(i)} = k_j^{(i-1)} \]  
(4.4)
for $j = 1, i - 1$ and

$$k_i^{(i)} = \frac{r_i + \sum_{j=1}^{i-1} a_{i-j}^{(i-1)} r_{i-j}}{E^{(i-1)}}$$

(4.5)

then we update the energy as follows

$$E^{(i)} = (1 - k_i^{(i)} k_i^{(i)}) E^{(i-1)}$$

(4.6)

The coefficient is then computed as

$$a_{i}^{(i)} = a_{i}^{(i-1)} - k_i a_{i-j}^{(i-1)}$$

(4.7)

for $j = 1, i - 1$ and

$$a_{i}^{(i)} = -k_i^{(i)}$$

(4.8)

Another LPC-based parametrization can be obtained using linear prediction cepstra, which can be more easily computed using a simple recursion as follows

$$c_n = -a_n + \sum_{i=1}^{n-1} (n - i) a_{i} c_{n-i}$$

(4.9)

### 4.3.2 Mel-frequency cepstral coefficients (MFCC)

Experimental evidence has shown that the human ear does not resolve frequencies linearly across the audio spectrum, and experiments suggest that using a similar model to encode a signal can greatly improve recognition performance. An alternative to LPC is a filter-bank analysis. This is somewhat similar to the result of a Fourier transform, but the main difference is that the frequency bands, instead of being linearly distributed over all frequencies, are positioned logarithmically, as shown in Figure 4.1. This logarithmic scale is called the mel scale, and is defined as

$$\text{Mel}(f) = 2595 \log_{10}(1 + \frac{f}{700}).$$

(4.10)

This filter-bank is implemented by taking the magnitude of the Fourier transformation for a window of speech data, and then multiplying the resulting magnitudes coefficients using each triangular filters. Thus, each bin will contain a weighted sum representing the spectral magnitude in the corresponding particular channel. This can be done also using the power in place of magnitude. Usually, we use the cepstral parameters (the famous MFCCs) that can be calculated from the log filter-bank amplitudes $m_j$ using the discrete cosine transform

$$c_i = \sqrt{\frac{2}{N}} \sum_{j=1}^{N} m_j \cos \left( \frac{\pi i (j - 0.5)}{N} \right)$$

(4.11)

### 4.3.3 Energy measures

We can add to the coefficients derived from MFC or LPC, an energy term which is the log of the signal energy, and for a signal $x_1 \ldots x_n$ is computed as follows

$$E = \log \sum_{n=1}^{N} x_n^2$$

(4.12)
4.3.4 Delta coefficients

The performance could be further improved when incorporating the time derivatives of the basic static parameters (the LPC or MFCC coefficients). The delta coefficients are computed using the following formula:

\[
d_t = \frac{\sum_{\theta=1}^{\Theta} \theta (c_{t+\theta} - c_{t-\theta})}{2 \sum_{\theta=1}^{\Theta} \theta^2}
\]  

(4.13)

where \(d_t\) is a delta coefficient at time \(t\), and \(\Theta\) is parameter specifying the time window over which these coefficients are calculated.

4.3.5 Acceleration coefficients

An additional parameter can be used, which is the acceleration of the delta coefficient. To obtain the acceleration coefficients, Equation 4.13 is applied to the delta coefficients.

4.4 Hidden Markov models (HMMs)

One might consider a bird song coded using one of these feature extraction methods at discrete time steps, as a sequence of observations, the feature vectors. We can assume such a sequence as being produced by a discrete-time dynamical system governed by a Markov chain. In the case of a regular Markov chain, each state emits an observation and we simply need to find the transition probabilities between states. Unfortunately, a model in which each state corresponds to an observable event is too restrictive to be applied in many problems, and usually (for example in case of bird songs) we have no idea of what these states are and what do they correspond to. Thus, we need to extend this model to allow for an observation being a probabilistic function of the state. This results in a hidden Markov model (HMM), which is a doubly embedded stochastic process where the state sequence is not observable (this is why it is called hidden), in which case the sequence of observations will not tell us what is the state sequence that generated them. The challenge is to estimate the parameters of the HMM (we call this process training) from sample observations, and then use the estimated parameters...
to infer the probability that a dynamical system could produce an observed sequence (a song produced by an unknown bird).

Mathematically, a HMM is a five-tuple \((\Omega_X, \Omega_O, A, B, \pi)\), where \(\Omega_X = [S_1..S_N]\) is a finite set of \(N\) distinct states, while \(\Omega_O = [v_1..v_k]\) is the set of possible observation symbols (the alphabet of observation), where \(\lambda = (A, B, \pi)\) denotes the parameters of the hidden Markov chain, with \(A_{N \times N}\) the transition probabilities matrix, \(B_{N \times k}\) the probabilities of observing each symbol for each state, and \(\pi_{1 \times N}\) the distribution of the initial state.

\[
A_{ij} \leq 1, \sum_{j=1}^{N} a_{ij} = 1 \tag{4.15}
\]

**Figure 4.2:** Structure of a hidden Markov model (HMM), where the states \(s_1..s_5\) are shown.

The state of system can change at discrete time steps, and the next state is defined by the matrix of transition probabilities \(A\). The state of the system at time \(t\) is denoted \(q_t\). The Markov chain assumption states the probability of passing from state \(S_i\) into state \(j\) at time \(t\) depends only on the current state only and not on the previous state changes, that is

\[
P(q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, \ldots) = P(q_t = S_j | q_{t-1} = S_i) = a_{ij} \tag{4.14}
\]

Also, we consider this probability independent of the time and \(a_{ij}\) is called the state transition (from state \(i\) to state \(j\)) coefficient \((A\) is the matrix composed of \(a_{ij}\) for all \(i, j\)), having the following properties:

\[
0 \leq a_{ij} \leq 1, \sum_{j=1}^{N} a_{ij} = 1 \tag{4.15}
\]
In the case where the process models a coin-tossing experiment, the alphabet of observation is simply $\Omega_O = \{\text{head, tail}\}$. But for bird songs, things become much more complex, and an example would help the reader to understand better what would $\Omega_O$ be in that particular case. Let us assume that we extract the dominant frequency from a song every 10 ms, we would end up having a sequence of observed values we denote $O = [O_1..O_T]$. The problem is that the observed values in $O$ are not discrete elements, thus $\Omega_O$ cannot be neither. A solution would be to use Vector Quantization (VQ), where we discretize the possible observations by saying that $v_1$ corresponds to a dominant frequency between 0 kHz and 1 kHz, $v_2$ between 1 kHz and 2 kHz, and so on... in which case the sequence of frequencies $[1022, 3500, 2422...]$ will be noted $O = [O_1 = v_1, O_2 = v_3, O_3 = v_2...]$. Such a solution is not convenient as subtle information may be lost during this process. The other solution, which we prefer as it does not suffer from such severe data loss, is to model the observation probabilities as mixtures of multi-dimensional Gaussian distributions, where the observation probability distribution for symbol $O_t$ while in the state $j$ is given by the value $b_i(O_t)$, and is defined as follows:

$$b_i(O_t) = P(O_t|q_t = S_j) = \sum_{m=1}^{M} c_{jm}N(O_t; \mu_{jm}, \sigma_{jm})$$ (4.16)

where $M$ is the number of mixtures, $c_{jm}$ is the weight of the $m$-th component. $N$ is the probability to generate $O_t$ following a Gaussian distribution with parameters $\mu_{jm}, \sigma_{jm}$.

In practice the parameters of HMMs are unknown and have to be estimated from data during the training phase. The choices made for these parameters are determinant for performance of the model and should reflect as reliably as possibly the data to be modeled.

### 4.4.1 The three problems of HMMs

As defined in (Rabiner, 1989), there are 3 majors problems to resolve in order to use the model in real-world applications:

**Problem 1** Given an observation sequence $O = O_1..O_T$ and a model $\lambda = (A, B, \pi)$, how to compute the probability $P(O|\lambda)$ that $O$ was produced by the model $\lambda$? In other terms, we are interested to score each model according to how well it matches the observation.

**Problem 2** Given an observation $O$ and a model $\lambda$, how to choose the ”correct” state sequence $Q = q_1..q_T$ that best explains the observation? Here, the problem is to attempt to discover the hidden part of the model.

**Problem 3** How to adjust the model parameters $\lambda = (A, B, \pi)$ in order to maximize $P(O|\lambda)$? This is a crucial aspect of HMMs as it is the training phase, i.e. optimally adapt the parameters of $\lambda$ to each sample in the training dataset.

### 4.4.2 Solutions to problem 1

A straightforward solution to this problem is simply to enumerate every possible sequence of length $T$ and compute their probability to be observed (assuming statistical independence...
between observations), that is

\[ P(O|Q, \lambda) = \prod_{t=1}^{T} P(O_t|q_t, \lambda) = b_{q_1}(O_1) \cdot \ldots \cdot b_{q_T}(O_T) \]  \hspace{1cm} (4.17)

Also, the probability for such a sequence to occur is given by

\[ P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \ldots a_{q_{T-1} q_T} \]  \hspace{1cm} (4.18)

The joint probability of \( O \) and \( Q \) occurring together is simply the product of Eq. 4.17 and 4.18, that is

\[ P(O, Q|\lambda) = P(O|Q, \lambda) \cdot P(Q|\lambda) \]  \hspace{1cm} (4.19)

Finally, the probability of the observation is simply a sum of this joint probability over all possible sequences of length \( T \)

\[ P(O|\lambda) = \sum_{all Q} P(O|Q, \lambda) \cdot P(Q|\lambda) \]  \hspace{1cm} (4.20)

\[ = \sum_{q_1 \ldots q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \ldots a_{q_{T-1} q_T} b_{q_T}(O_T) \]  \hspace{1cm} (4.21)

Unfortunately, this value cannot be computed using Eq. 4.20, as its computational complexity is \( O(2N^T) \) (\( N^T \) possible state sequences and \( 2T \) computations for each of them), so a more efficient method to compute \( P(O|Q) \) is needed. In the case of \( N = 5 \) states and \( T = 100 \) observations, \( 10^{72} \) computations are needed.

The Forward-Backward procedure

Consider the probability of the partial observation sequence \( O_1 \ldots O_t \) until time \( t \) and state \( S_i \) at time \( t \) under the model \( \lambda \), and we denote this value as being:

\[ \alpha_t(i) = P(O_1 O_2 \ldots O_t, q_t = S_i|\lambda) \]  \hspace{1cm} (4.22)

this value can be found inductively as follows:

1. Initialization

\[ \alpha_1(i) = \pi_i b_i(O_1) \]  \hspace{1cm} (4.23)

2. Induction

\[ \alpha_{t+1}(j) = \sum_{i=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \]  \hspace{1cm} (4.24)

3. Termination

\[ P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \]  \hspace{1cm} (4.25)

Using this other procedure, the complexity is reduced to \( O(N^2 T) \) rather than \( O(2TN^T) \), and in the case \( N = 5 \) and \( T = 100 \), only 3000 computations are needed, which is by far more tractable than \( 10^{72} \). Strictly speaking, the computation shown above is only the forward procedure, and the backward procedure is not needed to solve problem 1, but is used for problem 3, so I will not present it here.
4.4.3 Solution to problem 2

Here we need to find the best state sequence associated with the observations. Unfortunately, no exact path does exist as the states do not correspond to anything concrete. Thus only an optimal path according to certain criterion can be found. Once this criterion is chosen, the Viterbi procedure will be used to find the optimal state sequence. This can be seen as finding the optimal path in a lattice, where the $y$ axis corresponds to the states and the $x$ axis to time steps.

**Viterbi algorithm**

To find the best state sequence $Q = q_1 \ldots q_T$ for the given observation $O = O_1 \ldots O_T$, we need to define the quantity

$$\delta_t(i) = \max_{q_1, \ldots, q_T} P(q_1 \ldots q_T = i, O_1 \ldots O_T | \lambda)$$

which is the highest probability along a single path which accounts for the first $t$ observations and ends in state $S_j$. By induction we have

$$\delta_{t+1}(j) = \max_i \delta_t(i) a_{ij} b_j(O_{t+1})$$

To retrieve the whole state sequence, we use the array $\psi_t(j)$ to keep track of the argument maximizing Eq. 4.27 for each $t$ and $j$, and use the following procedure to retrieve the best state sequence

1. **Initialization**

   $$\delta_1(i) = \pi_i b_i(O_1)$$
   $$\psi_1(i) = 0$$

2. **Recursion**

   $$\delta_t(i) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(O_t)$$
   $$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}]$$

3. **Termination**

   $$p^* = \max_{1 \leq i \leq N} [\delta_T(i)]$$
   $$q^*_T = \arg \max_{1 \leq i \leq N} [\delta_T(i)]$$

   The optimal path (state sequence) is then given by

   $$q^*_t = \psi_{t+1}(q^*_{t+1})$$
4.4.4 Solution to problem 3

This is by far the most complex problem, which is how to adjust the parameters of the model \((A, B, \pi)\) in order to maximize the probability to observe a sample sequence, and can be viewed as the training procedure itself. If the state sequence were known it would be easy to estimate the parameters by collecting the data emitted in each state and then apply the EM algorithm.

Unfortunately the state sequence is hidden, so no optimal procedure to estimate the parameters of \(\lambda = (A, B, \pi)\) exists. Thus a method called the Baum-Welch procedure is used to estimate these parameters. This algorithm uses an initial guess of the parameters and then uses this guess to match the observation with the data generated with every possible state sequence to re-estimate the parameters. This is done for all samples until an optimality criterion is reached. However, description of this algorithm is beyond the purpose of this report, and thus will not be described here. The interested reader would find detailed information about this topic in (Rabiner, 1989; Young et al., 2002), where detailed explanation of HMM training procedures are given.

4.5 HMM Toolkit (HTK)

The Hidden Markov Model Toolkit (HTK)\(^1\) is a package of libraries and tools written in C at the Cambridge University that provides sophisticated facilities for designing, training and testing HMMs, as well as analyzing the results. For this project, we decided to use this software to implement the recognition of species and individuals, as it is one of the most used and flexible implementation of HMMs, and it is widely used in speech recognition. Another reason to use this software is that its source code is freely available, and can thus more easily be implemented on the embedded platform. Many other implementations exist in many other languages.

There are 6 important steps in order to use HTK efficiently:

- **Recording and labeling** In this step we must isolate and label the samples that will be used later to train the network.

- **Acoustical analysis** Now we need to extract the features from the signals, and to store them in .mfc files.

- **HMMs definition** We must design the structure of the individuals HMMs corresponding to each specific class, in our case bird species or individual.

- **Training** We will train the HMMs in order to adapt their parameters so as to best explain all the elements in the training set.

- **Recognition** Here we will finally feed an observation to the HMM, and we want to know which class it belongs to.

- **Performance evaluation** Now we finally analyze the recognition performance of the model.

\(^1\)Official HTK is [http://htk.eng.cam.ac.uk/](http://htk.eng.cam.ac.uk/).
Unfortunately, a certain amount of time is needed to perform all these steps manually, especially to create all the necessary configuration files and scripts. For this reason, I wrote a PERL script I called HTK Automation Tool (HAT) that performs all these tasks automatically (see Appendix A). This reduces considerably the amount of time needed to use HTK. Besides, HAT allows to perform batch experiments to test many different parameter combinations on the same data set. I included it as a part of my framework, so the models are created and trained offline on a desktop computer using a large database of samples, and only the trained models are uploaded onto the sensor nodes, where the recognition process is done in real-time on the detected bird songs.

However, integrating HTK in the framework in the way I did is obviously not a performant approach for sensor networks, as the previous layers (detection and beam-forming) need to output a wave file and run the recognizer through a command in the terminal. Many precious CPU cycles could be avoided if HTK could be fully integrated in our framework using all the interesting services offered by EmStar, as being triggered by the detection layer using StatusDevice, and then read the data directly from the ring buffer in the audio server using SensorDevice, thus avoiding audio data to be stored as wave files and then reopened again. Unfortunately, the license under which HTK is released does not allow users to redistribute the source code. This is contrary to the idea of EmStar who is released under GPL license, and thus we cannot integrate a modified version of HTK in the CVS tree where EmStar is contained. Nonetheless, I successfully ported HTK on the Stargate platform during this project. I ran a huge amount of tests using this tool for several bird species and individuals, and I was interested into understanding how recognition performance could be affected by several factors. All details concerning the experiments we performed using HTK can be found in the Chapter 6 of this thesis. For practical purposes, I run all these tests on a desktop computer.

4.6 Discussion

HTK is a very efficient and flexible implementation of HMMs. Many parameters can be configured, there are many of them and their relationship is highly complex. Also, there is no "optimal" combination of values for these parameters, and some of them might work well in the case of a bird species and not at all for another one.

The theory of HMMs is based on the Markov assumption, which requires that every observation is independent. This is obviously not the case with bird songs, and is even more incorrect the more the overlapping between observations is increased. In despite of this fact, HMMs are a very efficient method to model bird songs, and remain one of the most efficient tools for individual recognition, and even more for species.

Finally, the standard methods included with HTK are definitely not adapted for bird songs, as they are very generic and assume that the only relevant information is contained in the energy in frequency bands. It would be very appropriate to use features that allow a maximal separability between bird species.
This chapter describes the different methods I used for localization and separation of acoustic sources, and also how to create beam-formers that can be used to improve the quality of the recorded signals. These algorithms have been developed by our collaborators from the electrical engineering department at UCLA, and the description of these algorithms given in the next section is highly inspired by the original description of the algorithm, as given in (Chen et al., 2002a) and (Chen et al., 2005). However, the experiments and the discussion of the issues concerning this algorithm are the result of my personal work.

5.1 DOA estimation

For this project, localization of the vocalizing source is a vital element, as it allows automatic tracking of birds that facilitate analysis of their social behavior. Besides, knowledge of the direction of arrival of a sound signal, allows one to create directional filters that can be used to enhance the quality (the SNR) of the recorded sound emitted by a specific bird. Finally, it can be further used for automatic camera tracking, which remains a difficult problem in case of reverberation as a room or forest, where estimation might be biased because of multi-path effects.

When localizing birds, individuals are rarely close to the microphones, so it can be assumed that the sound source are in the far-field. Thus, I assume straight propagation of wavefronts, and also that the signal has same amplitude at all microphones. Both these assumptions are required for this algorithm in order to perform correctly.
The classic time difference of arrival (TDOA) using cross-correlation method works poorly when multiple sources are present simultaneously. Other systems based on the MUSIC algorithm requires a large number of microphones to work, implying huge memory and CPU resources, thus high energy consumption, which makes the system not suitable for embedded systems. For these reasons, the Approximate-Maximum Likelihood (AML) is used by the localization algorithm. Also, AML is optimal in the ML sense when the FFT window is large, and this is proved in (Chen et al., 2002a).

Let there be $M$ wideband sources, each at an angle $\theta_m$ from the array with the reference direction pointing to the east. The sensor array consists of $P$ distributed sensors, each at position $r_p = [x_p, y_p]$, and we assume that the microphones are omnidirectional and have identical responses. The array used is composed of 8 microphones placed on a circle of radius $r$ in order to maximize the symmetry of the system relative to the bearing of the source, as we do not have prior information on source location. The array center which is given by $r_c = \frac{1}{P} \sum_{p=1}^{P} r_p$, will be used as a reference point relative to which time-delays will be computed.

The relative time delay of the $m^{th}$ source is given by

$$t^{(m)}_{cp} = t^{(m)}_c - t^{(m)}_p = \frac{(x_c - x_p) \cos \theta_m + (y_c - y_p) \sin \theta_m}{v} \quad (5.1)$$

where $t^{(m)}_c$ and $t^{(m)}_p$ are the absolute time delays from the $m^{th}$ source to the centroid and to the $p^{th}$ sensor respectively, and $v$ is the speed of propagation of sound in air (we used $v = 345$ m/s). The data received by sensor $p$ at time $n$ can be modeled as:

$$x_p(n) = \sum_{m=1}^{M} S^{(m)}(n - t^{(m)}_{cp}) + w_p(n) \quad (5.2)$$

**Figure 5.1:** Placement of the sensors and sources.
for \( n = 1, \ldots, N \) where \( N \) is the length of the data vector, \( M \) is the number of sources present, \( S^{(m)} \) is the \( m^{th} \) source signal arriving at the array centroid position, and \( w_p \) is the noise represented by a zero mean white gaussian noise with a variance \( \sigma^2 \).

In order to simplify the derivation and analysis, the received signal is transformed into the frequency domain using a DFT. Afterwards, the array is steered to the estimated DOA so that signals are coherently combined for each frequency bin. For an \( N \)-point DFT, the array data is given by

\[
X(\omega_k) = D(\omega_k)S(\omega_k) + \rho(\omega_k), \forall k = 0, \ldots, N - 1,
\]

where the array data spectrum is

\[
X(\omega_k) = [X_1(\omega_k), \ldots, X_P(\omega_k)]^T,
\]

the steering matrix is

\[
D(\omega_k) = [d^{(1)}(\omega_k), \ldots, d^{(M)}(\omega_k)],
\]

the steering vector is

\[
d^{(m)}(\omega_k) = [d^{(m)}_1(\omega_k), \ldots, d^{(m)}_P(\omega_k)],
\]

and the source spectrum

\[
S(\omega_k) = [S^{(1)}(\omega_k), \ldots, S^{(M)}(\omega_k)]^T.
\]

The noise spectrum vector \( \rho(\omega_k) \) is a zero mean complex white Gaussian noise, with variance \( N\sigma^2 \). Superscript \( T \) denotes the transpose of a matrix and \( H \) denotes the complex conjugate.

The maximum likelihood estimation of the source DOA is given by

\[
\max_{\Theta, S} L(\Theta, S) = \min_{\Theta, S} \sum_{k=1}^{N/2} \|X(\omega_k) - D(\omega_k)S(\omega_k)\|^2
\]

which is equivalent to a nonlinear least squares problem. This equation can be solved using the technique of variables separation, by finding of \( \Theta \) that maximizes the following equation

\[
\max_{\Theta} J(\Theta) = \max_{\Theta} \sum_{k=1}^{N/2} tr(P(\omega_k, \Theta)R(\omega_k)),
\]

where

\[
P(\omega_k, \Theta) = D(\omega_k)D^\dagger(\omega_k),
\]

with

\[
D^\dagger(\omega_k) = (D(\omega_k)H\tilde{D}(\omega_k))^{-1}D(\omega_k)^H
\]

being the pseudo-inverse of the steering matrix \( D(\omega_k) \), and

\[
R(\omega_k) = X(\omega_k)X(\omega_k)^H
\]

is a snapshot of the covariance matrix.

This algorithm has been adapted to work with the data recorded using both the EE and the CENS array. However, the EE array is plugged directly on a computer because more microphones demand larger memory, faster CPU and higher energy consumption. At this stage of the project, we were more interested into identifying the main design parameters for the array and to quantify their influence on the global performance, this is the main reason why the algorithm has implemented only in Matlab, and I did not attempt to implement it on the embedded platform.
5.2 Array design parameters

The design of the sensor array, such as the number of sensors used and the intra-sensor spacing, has a direct influence upon the accuracy of the DOA estimation. The problem is that there is no magic array configuration that is optimal for all bird species, as the optimal value for these factors to allow maximal localization accuracy are a function of the dominant frequency of the signal to localize. To record a signal having a wavelength $\lambda$ with an array where the distance between microphones is $a$, two situations can be distinguished. On one hand, in the case where $a > \lambda/2$, grating lobes\(^1\) can appear in the beam pattern (this is the polar plot of function $J(\Theta)$, for $\Theta \in [1^\circ \ldots 360^\circ]$) and, in some cases, the height of the side lobes could have the same size, if not more compared to the main lobe. Such case could lead to a problem of ambiguity in the DOA estimation. On the other hand, when $a < \lambda/2$, no grating lobe appear, but the width of the main lobe increases, thus we are losing in angular resolution. A solution is to add more microphones, as having more microphones will decrease the width of the main lobe, even when the sensor spacing remains identical. This can be interpreted as adding more sampling points in order to decrease spatial aliasing. Localization performance is highly dependent upon the characteristics of the source spectrum, and performance can be optimized by designing an array customized for each particular source. As we want to localize multiple species, having very different songs, a trade-off between performance and generality for different birds is required. The interested reader is invited to consult (Chen et al., 2005), for a more detailed discussion concerning the optimal array size for several bird species that are of interest in this project.

I performed several simulations to show the influence of array size and number of sensors upon localization results. A single channel of recording was artificially delayed to emulate the other channels according to the structure of the array to emulate the source at a specific DOA, and the AML algorithm was applied to the resulting signals to localize the sound source. Such simulation, is the best case for the AML algorithm as the signals at different sensors is exactly the same, while in real experiment the microphones have slightly different gain and frequency responses, also the real time delays between sensor might differ from the array model. Figure 5.2 shows a polar plot of the value of $J$ in Eq. 5.9 for the case when 16 and 128 sensors are used and the distance between the sensors and center of the circle is 50 cm. The peak in the beam-pattern corresponds to the estimated DOA, and one can see that there is a clear unique peak at $0^\circ$, which is the estimated angle. However, the value of $J$ for all other angles is rather high and when fewer sensor are used, the value of $J$ has many other peaks for other than the real DOA. Figure 5.3 shows the same simulation, but in this case, $r = 500$ cm. Now the value of $J$ is much lower for all other angles, but still many other thin peaks can be seen when only 16 sensors are used.

5.3 Signal enhancement using beam-forming

Better recognition results could be obtained if the signals to test are good quality recordings with a high SNR, as parasitic noise overlapping in frequency with the signals of interest might be used by the HMMs, thus reducing the recognition performance. However, when using acoustic sensor networks, signals recorded at different locations can be recombined to reconstruct original vocalization and removing interferences at the same time.

\(^1\)local maxima in function $J$ for other $\Theta$ than the real DOA of the source.
5.3. Signal enhancement using beam-forming

Based on the DOA, we can use beam-forming to design spatial filters that will enhance signals coming from that specific direction, while attenuating signals emanating from any other direction. This process is very interesting for us, as it allows to improve the SNR significantly, as suggested in (Chen et al., 2005).

**Figure 5.2:** Simulated beam-pattern where channel 1 of a BAS was artificially delayed (radius $r = 50$ cm), source is at $0^\circ$. **Left:** 16 microphones. **Right:** 128 microphones.

**Figure 5.3:** Simulated beam-pattern where channel 1 of a BAS was artificially delayed (radius $r = 500$ cm), source is at $0^\circ$. **Left:** 16 microphones. **Right:** 128 microphones.
Beam-forming is done as follows: once we get the estimate $\Theta$ of the DOA using equation 5.9 for each source, the array is steered to the estimated DOA so that the signals coming from that direction are coherently combined for each frequency bin. The equation of this beam-former is given by

$$S_{ML}(\omega_k) = D^\dagger(\omega_k, \Theta)X(\omega_k)$$

where $S$ is the resulting signal in frequency domain, that can be transformed back to time domain using an inverse Fourier transform. Signal separation is performed using physical source separation where the SNR of each source is maximized in the ML sense. Note that in the case of multiple sources, the complexity of the AML algorithm is multidimensional, as opposed to MUSIC.

### 5.4 Localization

The exact location of the sound source can be obtained by combining the local DOA estimations of multiple microphone arrays, that have been previously translated into a global referential system. The exact location is then obtained by applying the least squares method to the intersections of the DOA estimates. However, due to lack of time, I did not focus on locating the exact position of a sound source, but only on DOA estimation. This aspect is discussed in (Wang et al., 2005a).

### 5.5 Discussion

The publications I mentioned that describe the AML algorithm, show that good source DOA estimates can be obtained when using simulation, and this is obvious due to the nature of the algorithm. In contrast, when using the real data recorded with the array, the array model has to correspond exactly to the real array, and the far-field assumption must be correct. If these assumptions are not correct, recombination of the signals might yield incorrect results. Besides, omnidirectional noise in real experiments might have a strong influence, as many side lobes will appear because the noise will be recombined and localized as well, in which case the beam-pattern will have a high value for all directions, resulting in ambiguous source DOA estimation.

Concerning the beam-forming, mainly simulations have been performed to show that source separation can be done correctly. Unfortunately, if the DOA estimates are incorrect, the beam-forming will not be done correctly neither. As I explained, simulation where a channel is shifted according to the array model is the optimal case as the DOA estimation is not be influenced by the omnidirectional noise. Beam-forming also uses a perfect model of the array, thus recombining signals recorded with an array that does not fit perfectly to the model, will result in suboptimal recombined signal.

Furthermore, as mentioned in (Chen et al., 2005), when beam-forming is used, the SNR of the resulting signal is increased by a factor $n$, where $n$ is the number of microphones. Unfortunately, a theoretical proof of this statement is missing and real experiments supporting it would have been more than welcome.
Also, another point is worth mentioned concerning this latter aspect. The AML algorithm itself is not able to tell with exactitude how many sources are present at a given time in the environment, as the user needs to have a prior idea about the upper bound of the number of sources that can be present at any time (let’s call this upper bound $n_{max}$). The AML will simply take the highest $n_{max}$ peaks in the beam-pattern, but they won’t all be relevant in every case, as when only one bird is singing other peaks might have been caused by the noise. Thus it would be appropriate to select only the peaks in the beam-pattern that have a value above a certain threshold. Unfortunately, as I will show later, when the signals used to localize are not bandpass filtered, the beam-pattern might have a value between 0.95 and 1 for all directions, which does not tell us much about the real location of the source.

The AML algorithm suffers obviously from these requirements that can be hardly meet with real arrays. This problem is even amplified as more microphones are added, as the small errors are multiplied. However, the main strength of this algorithm resides in its ability to cope with multiple sources overlapping in time and frequency, which is obviously a must in noisy environments such as rain forests.
In this chapter, I will present the results obtained using my framework, for each of the different procedures presented in the previous chapters. Afterwards, I will discuss these results and the factors that might affect the performance.

6.1 HHM recognition

This section describes several experiments I performed in order to quantify the performance of HMMs, and analyze what factors influence the performance. Afterwards, I’ll try to relate these factors with the song structure itself to show that better results could be obtained when taking into account relevant characteristics of bird songs. HMMs have been used in this project in two very different contexts that are both important in this project.

Antbirds species recognition In this context, I will focus on recognition of different antbird (AB) species based on their vocalization. Whole songs (duration approx. 2-3 sec.) have been cut manually from audio recordings made in Chajul reservation. I ended up using 25 samples for each of the six species of interest, and different amount of background noise and a high variability in song length have been selected. This choice is motivated by the fact that I wanted to minimize the possibility that the HMMs use properties of the noise instead of the bird song itself.
From this set, 15 samples have been used to train the HMMs, and 10 samples to test the recognition performance. The reason is that I do not have a large amount of recordings, and individual recognition is not feasible with the antbirds, due to the fact that when the songs have been recorded, there was rarely a way to ensure which individual produced the song. The birds in Chajul are not labeled, and can be rarely identified due to limited vision caused by the dense vegetation and other birds singing simultaneously.

<table>
<thead>
<tr>
<th>Species</th>
<th>Training samples</th>
<th>Testing samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>DWAW</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>DAB</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>BAS</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>GAS</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>MAT</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>DABF</td>
<td>15</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 6.1: The different species used to test the HMM performance, and number of samples used for training and testing for each of the species considered. The acronyms are described in Appendix B.

Acorn Woodpecker individual recognition. This other context is even more useful for the study of social interactions, as in the goal is to recognize individuals within acorn woodpeckers (ACW) groups. The calls of ACWs are fundamentally different from ABs, in many aspects. I was interested to show that vocal individuality exists in acorn woodpeckers and that HMMs can be used to classify individuals efficiently. For each individual, I used 100 samples of one of their particular vocalization composed of 2 syllables, called the \textit{waka} call. Some recording of other individuals were available, but I preferred not to use them as I had only few samples for the other individuals. Also, the waka calls are usually not isolated, and a bird uses usually several calls. This can be useful to recognize individuals, as if most calls in a sequence of calls can be recognized correctly, the caller might still be recognized regardless of several misclassified calls in the whole sequence. Additionally, localization can be used to ensure that all these calls have been emitted by the same individuals.

<table>
<thead>
<tr>
<th>Species</th>
<th>Training samples</th>
<th>Testing samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>2056</td>
<td>80</td>
<td>20</td>
</tr>
<tr>
<td>2428</td>
<td>80</td>
<td>20</td>
</tr>
<tr>
<td>3082</td>
<td>80</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 6.2: The different Acorn woodpeckers and the number of samples we used for each of them to test individual recognition performance using HTK.

Performance metric. For both contexts I have just defined, the performance metric used to compare the efficiency of the HMMs models is the one proposed in the HTK Book (Young et al., 2002). For each species or individual \(i\) tested, performance is measured simply as the ratio of
correctly recognized samples \((C_i)\) over the total amount of samples \((N_i)\) used for testing, and is defined as

\[
\text{Percent Correct for species } i = \frac{C_i}{N_i} \times 100\% \tag{6.1}
\]

The global performance of each experiment is then the average of the performance for each class \(i\). This metric is obviously not an absolute measure and is hardly representative of the general performance of the system for any data, as it only counts how many samples are correctly classified over the small data set at disposal.

For each experiment, I performed 50 runs using random repartition of the files into the training and testing sets, because the same experiment can have very different results, depending on which files are used in the training set and in the testing set.

### 6.1.1 Window size and overlapping

In the first experiment, I wanted to see how the size of the window and the amount of overlapping, as defined in Figure 4.2, affects the recognition performance. These parameters are set in the `config.mfc` file, which contains the parameters used by the feature extraction process.

\[
\begin{align*}
\text{WINDOWSIZE} &= 150000 \\
\text{TARGETRATE} &= 10000
\end{align*}
\]

The unit used is 100 ns, thus the window size is 15 ms, and target rate is 1 ms. I tried different combinations of these two parameters (see Table 6.3) to have a rough idea of their relative importance.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>WINDOWSIZE</th>
<th>TARGETRATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100000</td>
<td>10000</td>
</tr>
<tr>
<td>2</td>
<td>100000</td>
<td>50000</td>
</tr>
<tr>
<td>3</td>
<td>150000</td>
<td>50000</td>
</tr>
<tr>
<td>4</td>
<td>150000</td>
<td>120000</td>
</tr>
<tr>
<td>5</td>
<td>200000</td>
<td>50000</td>
</tr>
<tr>
<td>6</td>
<td>200000</td>
<td>150000</td>
</tr>
<tr>
<td>7</td>
<td>250000</td>
<td>50000</td>
</tr>
<tr>
<td>8</td>
<td>250000</td>
<td>150000</td>
</tr>
</tbody>
</table>

**Table 6.3:** The different values of the WINDOWSIZE and TARGETRATE I tested for the first experiment.

After running several experiments using different values of these parameters, I noticed that slightly better results (though statistically not significant in my opinion) are obtained when choosing a value of approximately 250000 and 150000.

One can see that the effect when using different combinations of these parameters more clearly in the case of ACW than AB. For the AB case, almost no difference can be seen, while these parameters have a much stronger impact in the ACW case. I suggest that the cause is more data is used in the individual recognition case thus subtle differences in performance can
be noticed better with larger databases. This is also supported by the larger variability in the ACW results.

6.1.2 Restriction of frequency range

For the second experiment, I wanted to show that limitation of the frequency range where feature extraction is performed has an effect upon performance. This limitation is actually used only by the MFCC method where the frequency bands are only distributed across this restricted range, instead of whole frequencies between 0 Hz up to the Nyquist frequency. The range parameters are set with the following lines in the config.mfc file.

```
LOFREQ=400
HIFREQ=7000
```

As one can expect, for both species and individual recognition, the performance of the system is slightly superior when frequency range is bounded according to the spectrum of the bird under consideration, and the reason is that the noise located outside this frequency range is not taken into account by the HMMs. Again, I ran 50 experiments using the method MFCC_E_D_A (see next section), with WINDOWSIZE=250000 and TARGETRATE=150000. In the case of antbirds, I have found an average performance of 88.4% with unrestricted frequency range, and 95.5% when I consider only frequencies in the range [800; 4000] Hz. For the acorn woodpeckers, I used the range [400; 7000] Hz, and the results are 89.4% when range is limited, and 85.6% if all frequencies are used.

6.1.3 Feature extraction methods

For this third experiment, I have been interested in testing the different standard feature extraction methods that comes with HTK. Mainly, I compared the two most commonly used methods in HTK, which are MFCC and LPC, and additionally I tested adding other basic features such as energy, delta coefficients, and acceleration of the features, in order to
6.1. HHM recognition

quantify how these parameters affect recognition. I decided to do this experiment by setting WINDOWSIZE=250000, and by varying the TARGETRATE parameter in the range [10000; 240000] for each of the different methods described in Table 6.4.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Method used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MFCC</td>
</tr>
<tr>
<td>2</td>
<td>MFCC_E</td>
</tr>
<tr>
<td>3</td>
<td>MFCC_D</td>
</tr>
<tr>
<td>4</td>
<td>MFCC_D_A</td>
</tr>
<tr>
<td>5</td>
<td>MFCC_E_D_A</td>
</tr>
</tbody>
</table>

Table 6.4: The different methods I used to test the performance of HTK. The characters after the underscore adds optional features to be also used for HMM recognition E stands for energy, D stands for delta coefficients, and A stands for Acceleration.

![Figure 6.2: Recognition performance for the values presented in Table 6.4. Left: results of species recognition with the antbirds. Right: individual recognition for the acorn woodpeckers.](image)

One can see that again, in the AB case the difference between different methods is hardly noticeable, and is rather stable between 92% and 95% over the whole range of values used for TARGETRATE. Note that 95% of correct classification means the 3 songs are attributed to the wrong species, which is a quite respectable results when taking into account that only 15 samples of different length and background noise levels are used to train the models. The reason might be the same as in the first experiment, but I will propose another hypothesis here. As the structure of the songs is very different across species, the samples of each species are very clustered around different positions in the feature space, with a high separability between the
clusters, even the parameters are estimated very roughly. Thus, changing the value of single parameter is not likely to have a large impact on the separability between these clusters. Also, adding other features such as the delta and the acceleration to the classification will not add discriminative power, as these features are very likely to be anyway very different between species, and does not bring much information to the classifier, but only increase the probability of correct classification. This hypothesis is supported by the quasi-perfect recognition of species which shows the great separability between classes, and also when I listened to the misclassified songs, their SNR is very low and I hardly perceived any bird song in these samples, thus it is possible that the HMMs used the noise in the signal rather than the song itself to classify.

Now in the case of individuals of the same species, the structure is very similar, and the separability between clusters is very small. In this case, changing slightly the value of a parameter might have a great impact on the separability between different classes, and adding other features will increase the discriminative power and thus the separability between the classes. It interesting to point out that adding the deltas \( \Delta \) improves the performance in the case of ACW, and the acceleration \( A \) improves it even slightly more, while the energy does not seem to have an influence \( E \), and this might give us some information about what the way individuality is encoded in ACW waka calls. Also, when the separability between classes is smaller it becomes more important to estimate correctly the parameters of the HMMs to allow accurate classification, thus much more training examples are required.

Finally, the vertiginous drop of performance that can be seen for low values of \( \text{TARGETRATE} \) shows one of the main weakness of HMMs. Decreasing this parameters for a constant window size will increase the overlapping between consecutive observation, thus also the correlation between them. But, the Markov models are based on the Markov assumption where each observation is statistically independent from the previous one. Unfortunately, with real signals as human speech or bird songs, even without overlapping between the samples used to extract the observation vectors, there is still a correlation between consecutive samples. However, we have shown that even when the Markov assumption is violated, classification with HMMs can be successful as long as the separability is sufficiently high.

I also ran several experiments in order to estimate the performance of LPC methods, and I found out that these methods yield a correct percentage approximately twice as low as the equivalent MFCC method, thus I decided to omit them from these results. I suggest that LPC is not appropriate for bird song modeling, as it is based on more complex information as opposed to energy in different frequency ranges. By no means, I suggest that MFCC is a biologically relevant and accurate model of the animal auditory system, but as mentioned before, many studies have shown that the dominant frequency might be one of the most efficient acoustic cue used by animals in individual identification, and this could explain the impressive results we got so far. Besides, in contrast to linear prediction coefficients, it is easier for the bird brain to extract frequency information from a signal, where neural pathways can act as delay lines.

### 6.1.4 Discussion about HMMs

I have shown that very good species recognition performance, over 95% of correct classification when 6 species are considered, can be obtained when using HMMs, even when very few samples (15 songs) are used to train the models. Besides, many of the samples used have a low SNR, where the bird song can hardly be distinguished from the background noise. Such promising results could be explained by the fact that vocalizations are very different across
species, thus are more clustered in the feature space. However, we noticed that some aspects complicate the species recognition task, as for example when a bird sings only a part of his song. If the songs differ greatly in terms of length across species, then increasing the state length (increase the \texttt{TARGETRATE} parameter, so a state is responsible for a longer segment of the song) is a considerable thing to do, as the resulting state transition probabilities will be more different across models, resulting in less ambiguous recognition.

It should be noted that the results presented here should not be taken as a comprehensive analysis of HTK performance, but rather as a simple indication of how recognition might be affected by several factors. The very restricted amount of data I used to test species recognition (25 samples per species) is very small and statistically irrelevant of the real influence of these parameters upon global performance. To have a more precise idea of the performances of our system in real field experiments, much more extensive tests should be performed, using a larger audio database with very different amount and nature of background noise.

I found that individual recognition is by far a more delicate process than species recognition. In the best cases, I had around 90% of correct classification. This is due to the fact the structure of the calls among individuals are very similar, and the features extracted from the signals might not be the ones that convey individual identity. I want to point out that when I used only 10 samples per individual to test the models as did for the experiments with the ABs, I had over 97% of correct classification. Also, I noticed that much more examples are needed to train models in the case of individual recognition, as the differences between songs are more subtle, and more data is required to estimate these slight differences correctly. In order to reduce the variability encountered in the experimental results, one might also want to have a bigger overlapping when extracting features, as this will increase the correlation among states, which is usually the case in the different song segments. This will allow to be less sensitive to the noise that is the main source of error in recognition.

It is very interesting to point out that ACW 3983 has been often misclassified as 3982\footnote{Result not presented in this report.}, and the person who performed the ACW recordings is able to distinguish easily individuals only by listening to their songs, but can hardly distinguish between 3982 and 3983\footnote{Source: Yuan Yao, personal communication.}, exactly as the HMMs did. The reason is that the former is the parent of the latter, and such a high similarity could be explained by the fact that siblings learn their songs by imitating the songs of their parents, though my experiments do not provide results that could support this hypothesis.

One parameter is worth being discussed, is the number of states used to model each species. It is difficult, if not impossible, to relate this parameter with any physical property of bird songs. I used a value of 5 states for ABs and 7 for ACW, as I just found that these values work best in average, without any other motivation. At first, I thought that adding more states will results in higher performance, given that a bird song is at least 2 seconds long and feature vectors are extracted every 10 ms. However, my hypothesis turned out to be incorrect, as performance did not change in a significant manner when using between 5 and 15 states, and adding more than 15 states even degraded drastically the performance. Even if optimal values for state number might differ for the models of the different species, using these values for each model is not suggested, as changing the properties of the model for one species affects the performance of the other species as well as it might be misclassified due to this change. The reason is that the
more states are added and the more parameters are needed to be estimated correctly, and much more training samples would be needed.

At this point of the discussion, I suggest that using the standard feature extraction proposed by HTK is a very generic procedure and might not reflect at all the features that are relevant and used by birds for identification of callers. If one could design a feature extraction module based only on relevant features that are different across species, performance could be significantly improved. I hope data mining techniques to give us informations about what these relevant features are, and based on this information, much more appropriate feature extraction methods, that look only for the cues that differ significantly across individuals, could be devised. Finally, I suggest that using better quality recordings for training and testing could significantly influence the performance, as noise has a strong impact on the global performance.

6.2 Field experiment

In this experiment, I wanted to compare the two microphone arrays described in section 3.1.3 and 3.2.1, by playing jungle recordings of four different antbird species, each through a loudspeaker located at a different position. For this purpose I created four wave files of a duration of 23 seconds, each one containing vocalizations of a single antbird. These files have been filtered to remove jungle noise (because noise in the jungle comes from all directions, not from a single one), and the spectrograms of these files is shown in Figure 6.4. I ran this experiment on the UCLA campus on a sunny and noisy (wind and people passing by and talking) sunday afternoon. The two arrays were placed next to each other and the four loudspeakers were placed at respective bearing of $\theta = 0^\circ$, $15^\circ$, $22.5^\circ$ and $45^\circ$ relative to west direction, at a distance $D$ from the arrays (see satellite image in Figure 6.3). I ran two experiments where the speakers with respectively $D = 10$ and $25$ meters as I wanted to quantify the influence of distance upon the azimuth estimation results. As I also wanted to evaluate how noise affects the performance of the system, I performed each experiment twice, one with the ambient noise only, and the second I added artificial jungle noise played through a speaker placed under the arrays. The nodes have been positioned using a laser range-finder for measuring the distances and a compass to measure the angles. Afterwards, I took GPS measurements of the exact locations using a Trimble GeoXH GPS handheld, and used the satellite data to measure the exact angles and distances between the points, which are given in Table 6.5. After running the experiment, I noticed that the arrays were not exactly perpendicular to the $0^\circ$ line, but rather they had a slight offset of approximately $3^\circ$ to the left, which makes the $0^\circ$ loudspeaker be at approximately $3^\circ$ on the right of the array.

6.2.1 Songs detection

I ran my detection algorithm over the file recorded using the CENS array where $D = 10$ and with jungle noise added (file used is /Starecs/exp10m_5.wav, its spectrogram is shown in Figure 6.5), and the result of the detection is shown in Figure 6.6. The parameters are $\alpha = 0$, $l_{\text{song}} = 0.794$ s, $l_{\text{min}} = 35$ ms, $l_{\text{sil}} = 35$ ms. Noise is initialized over the first 20000 samples of each recording. One can see that the four different bird songs have been successfully detected when they are not overlapping, but I had to use a lower value of $\beta$ (1.5 instead of 3) because

---

3I decided to use four CENS stargate platforms as loudspeakers so I can start the playback simultaneously from a laptop. However, as explained previously the quality of these loudspeakers is very poor, as is their amplifier. However, noise is exactly what I want, as the jungle is not a lab.
### Table 6.5:

For each location of the speakers, the real angles and distances as calculated using the GPS data (accuracy within 10 cm) is given, along with the wave file that was played from that location (see in Figure 6.4 the spectrograms of the wave files). The distances and angles are calculated relatively to the location of the sensors and do not take into account the orientation of the sensor arrays.

<table>
<thead>
<tr>
<th>Distance (m)</th>
<th>Angle (deg)</th>
<th>True angle (GPS)</th>
<th>True distance (GPS)</th>
<th>File played</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 m</td>
<td>0°</td>
<td>0°</td>
<td>No data</td>
<td>1.wav</td>
</tr>
<tr>
<td>10 m</td>
<td>15°</td>
<td>14.17°</td>
<td>10.03 m</td>
<td>2.wav</td>
</tr>
<tr>
<td>10 m</td>
<td>22.5°</td>
<td>22.41°</td>
<td>9.98 m</td>
<td>3.wav</td>
</tr>
<tr>
<td>25 m</td>
<td>0°</td>
<td>0°</td>
<td>24.93 m</td>
<td>1.wav</td>
</tr>
<tr>
<td>25 m</td>
<td>15°</td>
<td>16°</td>
<td>24.93 m</td>
<td>2.wav</td>
</tr>
<tr>
<td>25 m</td>
<td>22.5°</td>
<td>22.42°</td>
<td>24.89 m</td>
<td>3.wav</td>
</tr>
<tr>
<td>25 m</td>
<td>45°</td>
<td>44.7°</td>
<td>25.09 m</td>
<td>4.wav</td>
</tr>
</tbody>
</table>

### Figure 6.3:

Satellite image of the experimental setup. Four speakers are placed at different locations but same distance $D = 10$ and $D = 25$ from the microphone arrays, and at respective angles of $\theta = 0^\circ, 15^\circ, 22.5^\circ$ and $45^\circ$ with respect to the west direction.

The third and fourth song have a very low energy (see Figure 6.7). This approach should not be used in environments where the noise amount is important and highly variable, as it might
Figure 6.4: Spectrograms of the four recordings we prepared for this experiment, where a) is file 1.wav with DAB vocalizations, b) is file 2.wav with MAT, c) is file 3.wav with DWAW, and d) is file 4.wav with BAS. These acronyms are explained in Appendix B.
Figure 6.5: Spectrogram of the signal recorded with the CENS microphone array connected to a stargate node (only the first channel is shown). One can see that there is a rather large amount of noise, especially in the low frequencies under 500 Hz due to the wind and to the artificial noise added. However we can see clearly the vocalizations of the 4 birds. We might also notice that they have also a different intensity, that could be the case if some birds have fainter calls than others, and also for the birds being at different distances. To see how our system would perform in this case, we we did not normalize the intensity of files we play-backed.

We can notice that the algorithm detects the vocalizations as a single vocalization, as one might expect as it uses only the energy in the signal as criterion. I suggest that performing detection using beam-formed signals should yield much better results. Besides, I ran many other detection experiments over long recordings done in the jungle using a directional microphone (I do not show these results here), and when the noise level in this recordings is constant and the songs are higher than the noise, I got usually over 90% of songs in the recordings that are detected. This score is an average value estimated from my results over many experiments with different files recorded in Chajul. However, the optimal values for the parameters of the detection algorithm to recognize most of the songs in the recording, are different for each file.

### 6.2.2 Comparison of signal-to-noise ratio

For this second experiment, I wanted to compare the SNR of the same sound as recorded by the two arrays, and then to see experimentally whether the SNR is improved when beam-forming is applied. I played a MAT song through a loudspeaker located in front (bearing of 0°) of the arrays at respective distances of 5, 10, 15, 20, 25 and 30 meters. Half of each recorded file (samples $x_1..x_N$) contains ambient noise, and the other half (samples $x_{N+1}..x_L$) the signal, and I compute the SNR ratio for each array on a single channel by computing the power in the noise segment ($P_{\text{noise}} = \frac{1}{N} \sum_{i=1}^{N} x_i^2$), and the power in the song [$P_{\text{noise+signal}} = \frac{1}{L-N} \sum_{j=N+1}^{L} x_j^2$].
Finally, the SNR in decibels is estimated as follows:

$$SNR = 10 \log_{10} \frac{P_{noise+signal} - P_{noise}}{P_{noise}}$$  (6.2)

Afterwards, I recomputed the SNR for the same segments, but using a bandpass filter on the recorded files to filter out frequencies that are not part of the bird song (the filter is the same as used in Figure 6.6). Finally, I used beam-forming to improve the quality of the signal and recomputed the SNR again. The value of the SNR according to the distance between the source and the arrays is shown for these cases in Figure 6.8.

One can see, that a simple bandpass filtering on the signal already improves greatly the SNR, which is the result expected, as most of the noise is removed through this process. One should not take this graph as an absolute and accurate measure of the recording quality of both arrays, as the SNR values are simple estimates from a single recording, and the background noise is not constant in all of these recordings. However, in all these experiments, the SNR of the data recorded with the CENS array is consistently slightly better than the data recorded with the EE array, and I suggest that the cause for these differences are due to the different microphones used in the arrays.

It is astonishing to see that beam-forming did not improve the quality of the received signals as we expected. The SNR of the beam-formed is even lower than with the original sound.
6.2. Field experiment

Figure 6.7: The power spectral density (PSD) of the signal shown in Figure 6.6 as calculated by the CFAR algorithm. The horizontal lines show the average of the noise PSD ($\mu$), and the corresponding standard deviations of the noise ($\mu + \sigma, \mu - \sigma$). The thick clear line ($\mu + \beta\sigma$, here $\beta = 1.5$) is the threshold above which we consider the segment as possibly being a vocalization.

file, especially when beam-forming has been applied to the data collected with the EE array. However, I beam-formed the data according to the real DOA of the sources (0°), and I noticed that when using incorrect DOAs, the SNR was even worse, which suggest that SNR of the beamformed signals is maximal when the correct angle is used, but also when the assumptions needed by the AML algorithm are valid. Due to lack of time, we did not perform more complete experiments concerning beam-forming.

Unfortunately, beam-forming cannot be tested to reduce SNR in simulation where a single channel of a noisy recording is artificially delayed, and the reason is obvious, as the noise is shifted at the same time as the signal emulating that the background noise also comes from the same direction as the song, whereas ambient noise usually comes from all directions. However, simulation can still be used to separate between two sources as long as the different signals are correctly delayed according to their respective DOA, as has been done in (Chen et al., 2005).

Efficient beamforming is a capital process, and it will allow to increase greatly the quality of the recorded bird songs which is very likely to improve the performance of the species/individual recognition by the HMMs. As this process uses the same array as the AML localization we
Figure 6.8: Comparison of the Signal-to-Noise ratio estimates for the two microphone arrays used in this project. The dashed line represents the data recorded with channel 1 of the EE array, while the filled line is channel 1 from the CENS array. Note how only bandpass filtering greatly improves the SNR.

must that this model corresponds exactly to the real array, else the recombination of the signals will be incorrect.

6.2.3 Direction of arrival estimation

The results of the DOA estimation using data recorded with the two arrays are shown in Table 6.6. One can see that very interesting results have been obtained using the data collected with the CENS array, even when the sources were at 25 meters and with jungle noise added. Unfortunately, the results obtained using exactly the same code for the data collected with the EE array, are very random and incorrect. I did not figure out the exact reason why this happened, especially as the DOA estimated using the CENS array is close to the real DOA (with a small amount of error), and it is expected that the EE array does at least as well. One explanation for this behavior, is that the AML algorithm is based on how the experimental data corresponds exactly to the model of the array. As the algorithm expects the model of the array to represent exactly the real array, small errors in microphone placement the differ from the model might have a large impact upon performance.
### Table 6.6

The results of the DOA estimation using the AML algorithm. Column 2 and 3 are the angles estimated using the CENS 4 microphone array, while columns 4 and 5 show the angles estimated using the data recorded with the EE 8 microphone array.

<table>
<thead>
<tr>
<th>Point</th>
<th>CENS</th>
<th>CENS, noise</th>
<th>EE</th>
<th>EE, noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 m, 0°</td>
<td>5°</td>
<td>4°</td>
<td>4°</td>
<td>50°</td>
</tr>
<tr>
<td>10 m, 15°</td>
<td>15°</td>
<td>18°</td>
<td>1°</td>
<td>38°</td>
</tr>
<tr>
<td>10 m, 22.5°</td>
<td>27°</td>
<td>27°</td>
<td>36°</td>
<td>19°</td>
</tr>
<tr>
<td>10 m, 45°</td>
<td>50°</td>
<td>46°</td>
<td>54°</td>
<td>55°</td>
</tr>
<tr>
<td>25 m, 0°</td>
<td>12°</td>
<td>9°</td>
<td>−22°</td>
<td>−17°</td>
</tr>
<tr>
<td>25 m, 15°</td>
<td>24°</td>
<td>19°</td>
<td>34°</td>
<td>38°</td>
</tr>
<tr>
<td>25 m, 22.5°</td>
<td>32°</td>
<td>27°</td>
<td>46°</td>
<td>38°</td>
</tr>
<tr>
<td>25 m, 45°</td>
<td>55°</td>
<td>51°</td>
<td>−62°</td>
<td>55°</td>
</tr>
</tbody>
</table>

**Figure 6.9:** Beam-pattern of a BAS recorded with the CENS Array (radius $r = 5.66$ cm), source is at 45°. **Left:** simulated beam-pattern with delayed channel 1. **Right:** real beam-pattern using all 4 channels (DOA estimation = 50°).

Especially, the algorithm assumes that all microphones have identical gain, and that the far-field assumption is correct. This statement is supported by the simulated beam-pattern shown on the left in Figure 6.10, where localization has been performed using the channel 1 of the recording which has been artificially shifted according to the model, in order to emulate a signal coming from a specific direction. This also shows that my implementation of the algorithm is correct.

After the experiment, I noticed that one of the channels of the EE recordings (ch. 2) was dead as no data has been collected from this channel. I decided to use only four channels (ch. 1, 3, 5 and 7) of the eight channels. Even in this case, no better DOA results have been obtained,
so I think another problem caused the incoherent results I obtained. Another hypothesis is that the channels from the wave file have permuted and do not correspond to their location in the model, which I doubt being the cause, as different DOA estimates are obtained for the different recordings of the same source at the same location. On one hand, it is clear that ambient noise affects greatly the performance of the localization by creating side lobes in the beam-pattern of the recorded signal, and in most of the cases the beam-pattern was very high and similar in all directions, which is the most likely cause of these errors. On the other hand, in some cases the results of the estimation are very close to the real DOA.

**HMMs species recognition with field experiment data** HMMs recognition has also been applied to the birds detected, and in most of the cases the DAB, MAT, DWAW were always correctly classified, while the BAS were almost always classified as a DWAW. This is a very unexpected result, as the vocalization of the BAS is significantly lower in frequency as opposed to the DWAW. I suppose that the error was due to the very weak energy in the BAS signal as opposed to the other songs, and the noise is used by the HMMs for the classification rather than the signal.

Again, the last segment of the recordings where all songs are overlapped is usually detected as a DWAW or BAS, but it is evident that the HMMs are not able to separate the four songs and analyze them individually. Source separation followed by beam-forming, could separate the four songs, and classify them correctly. Unfortunately, lack of time prevented us to make more comprehensive analysis of the HMMs performance on the data collected in this experiment.
6.2. Field experiment

6.2.4 Discussion about AML

We have shown that relatively accurate DOA estimation can be obtained, even when only four microphones are used. Unfortunately, due to some unidentified technical errors, I have not been able to get sufficiently accurate estimation when using the EE array. However, I pointed out that the AML algorithm requires many important assumptions that are hardly guaranteed when using a real microphone array. The classic DOA estimation method using cross-correlation of signals, is less sensitive to errors in the model and might thus provide equivalent results, and also is simpler to implement on embedded platforms. This approach has been used with impressive results in (Girod, 2005b), but in this case the signals used for localization are especially designed for this purpose. Unfortunately, this method is not able to distinguish between different sources overlapping in time or in frequency, which is a common situation in field recordings. Henceforth, this method cannot be used in our project for other purpose than sensor nodes self-calibration process.

During my experiments, I noticed that the length of song segment used to localize the source and also the position where this segment is taken, can influence significantly the DOA estimation results. The reason is mainly because the background noise might be used by the AML algorithm to derive the angle of arrival of a source, and the noise amount varies during the song. Besides, I noticed that simple band-pass filtering of the signal as I did for the detection improves greatly the performances, and also was confirmed by specialists 4. When I used the raw signal for localization, the resulting beam-pattern was almost maximal for all directions, and distinct peak could be discerned, in which case the maximal value can be due to the noise, rather than the sound source itself. This is due to the fact that a large frequency range is used for localizing the source, even if the AML algorithm uses only the 50 frequency bins (resulting from the frequency domain transform of the recorded signals) that contain the highest energy. A more efficient approach would be take exclusively the frequency bins that contain the vocalization to be localized, which is unfortunately not known in advance.

At the time of writing of this report, our collaborators are currently using two EE microphone arrays on the Chajul site (See Appendix C), in order to test localization and beam-forming on recordings performed in the rain-forest. Unfortunately, the results will not be included in this report, as the data will be available only after when this report is due. It will be very interesting to see the results of localization in the jungle, as the large amount of noise coming from all directions, might have a large impact on the performance.

4Source: Chiao-En Chen, personal communication.
Chapter 7

Conclusions and future work

"It is not only fine feathers that make fine birds."

Aesop (620 BC - 560 BC)

This report has shown the current status of an interdisciplinary project in which we plan to design an autonomous and distributed sensing system able to perform collaborative analysis of bird songs in their natural habitats. To reach this goal, collaboration among various scientific domains is required, thus multiplying the problems that need to be solved. However, in despite of the complexity of the different components of my framework, I have shown that using tools primarily designed for human speech recognition such as HMMs, can be used for bird species and individuals recognition with very promising results. My experiments also support the hypothesis that vocal individuality exists in Acorn Woodpeckers and that their songs encodes information about the identity of the bird.

Besides, I have presented an algorithm that allows to detect efficiently segments which are likely to contain bird songs in continuous audio streams. This algorithm is very simple, thus appropriate to be implemented on sensor nodes, but the performance degrades when the amount of background noise in the environment is highly variable, as is the case in rain forests. Unfortunately, it lacks the flexibility offered by unsupervised machine learning methods, and it uses only very little of the information contained in a signal to infer the nature of the sound source. I showed that inclusion of several other criteria that are specific to bird songs could improve drastically the performance of this algorithm. These criteria can be derived using data mining techniques on large audio databases, and the knowledge of what cues allow a maximal separability between individuals, could be used to design appropriate feature extraction methods as front-end for HMMs, which would result in an optimal recognition performance.

Finally, I have discussed the AML algorithm that is used to localize birds. As opposed to classic cross-correlation TDOA estimation methods, AML is able to deal with multiple sound
sources overlapping in time and frequency. Unfortunately, some assumptions are vital to derive correct localization, and it is quite difficult to ensure the validity of these assumptions. I showed that relatively accurate localization can be done using data recorded with sensor arrays that do not respect these assumption. This is very encouraging, as it suggest that much better results could be obtained if using much better quality sensor arrays. Also, I discussed the beam-forming procedure, which able to improve the SNR ratio of the recorded signals, and thus performance of the recognition process. I pointed out the main weakness of the AML algorithm and proposed solutions to improve the localization results.

It should be pointed out that the sensor nodes used during this project are only first generation generic prototypes not designed to perform distributed real-time audio processing, nor are the microphone arrays adapted to bird songs spectral characteristics. New technologies with a reduced manufacturing cost and energy consumption, and in the meanwhile increased computational resources and storage, along with higher bandwidth communication, would allow the development of more appropriate sensor nodes to be used in field experiments and the implementation of much more performant localization and recognition methods.

The software can be greatly improved, and a single platform-independent framework written in C instead of Matlab, would be highly appropriate so that a single implementation of the algorithms needs to be coded. Development of the audio server concept based on the Sensor-Device pattern in EmStar, could allow the same algorithm to be applied to any audio stream regardless of how the data has been recorded. Further perspectives on this project include improvement of the performance of the different algorithms used, and a better interaction between the different processing stages, helped by a modular approach. Performance could be further improved using an appropriate design of the microphone array that would take into account the characteristics of bird songs. Integration of a DSP pre-processor to perform the signal processing, as is filtering, would allow the main CPU to be used for more critical tasks. Finally, a sound-card that allows direct memory access (DMA) is needed, as many precious CPU cycles are wasted when data is copied from the audio hardware into the user space.

I suggest that EmStar is an appropriate software environment to use for the design of such applications using sensor networks, as it contains several primitives that are highly desirable in the context of acoustic source localization. The design of EmStar allows to create robust and efficient modular software, unfortunately, the problem of power management has not yet been addressed explicitly. However, given that technology is constantly evolving, it makes more sense to focus on improving the performance of the algorithms used, rather than trying to optimize an algorithm for a given hardware platform. Energy is one of the most important bottlenecks that prevents long-lived deployments in remote environments, thus alternative power sources along with emphasis on careful power management are required.

In the long run, we could focus on the development of swarm-intelligent algorithms that can take full advantage on the distributed aspect of sensor networks in order to provide higher robustness and scalability to the system by reducing the complexity of the algorithms used. Also, we would like to investigate adaptive communication protocols that grounds sensor data into concepts which are shared via a symbolic adaptive language, that minimizes the data to transmit according to changes in the environment, exactly as birds do! But we first need to know how they do it, thus understanding of the neural mechanisms involved into human and animal language acquisition, storage, and recognition would be a considerable milestone for the whole scientific community.
Appendix A: HTK automation toolkit (HAT)

What is HAT

HAT stands for HTK Automation Toolkit. This is a PERL script I wrote for this project and its goal is make happy many guys that have to use HTK to perform sound recognition, but who don’t want to spend hours in order to read an HTK tutorial to understand how to use it. The purpose of HAT is to automatically generate all the scripts needed, create the HMM models, initialize them, recognize and finally analyze the results. The only requirements to use it is to have a correct installation of PERL and of HTK.

What do I need to do?

First, unzip the tar in a directory (let’s say /root/htk/). Then, you need to define a project to work on, for that create a new folder (let’s say /root/htk/birds) in which you will create a file called labels.hat that will contain a line for each model to build (thing to recognize, species, individual, etc) having the following syntax:

```
LABEL SAMPLES TEST STATES
```

where

- **LABEL** is the name of the species/sound/individual whatever to be recognized,
- **SAMPLES** is the number of samples available for this species,
- **TEST** specifies how many samples will be used to test the trained network (so SAMPLES-TEST samples are used to train it),
- **STATES** is the number of states in the corresponding markov model.

Here’s a sample labels.hat where we want to create 4 models corresponding to 4 species of birds:

```
wrench 25 5 13
pigeon 25 5 22
finch 25 5 21
eagle 25 5 16
```
The final thing you need to do is to provide the sound files (we assume that the data is in wave file format). Create a folder "/root/htk/data/snd" and place the sound files in this folder. The final requirement is that their filename is exactly the same as the labels we put in the labels.hla file, followed by the number of the sample. So we’ll have the files wrench-1.wav to wrench-25.wav, then pigeon-1.wav to pigeon-25.wav and so on. From these 25 files, 5 files will be used in order to test the model, and the 20 others will be used to train the models.

Now just go into the directory /root/htk/ and run the following command in the terminal

```
user@world$ perl hat.pl project
```

where project is the folder in which the project you want to use is contained (in our example it is birds), then sit down and relax.

The script is fairly well commented, so feel free to have a look at it if you are courageous, and modify it for your personal usage, especially if you want to change the parameters of the feature extraction process.
Appendix B : Features

This appendix presents the acoustic features that can be measured using the Soundruler software. Source: Soundruler official website http://soundruler.sourceforge.net.

Note: 0% actually refers to the edge of the call, which is defined as when the algorithm from the peak finds an amplitude drop to 1% or an inflection after 10%.

### Amplitude

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PulsePeak</td>
<td>Maximum amplitude of the pulse</td>
</tr>
<tr>
<td>Pos_0_Beg</td>
<td>Position at initial 0% peak amplitude</td>
</tr>
<tr>
<td>Pos_10_Beg</td>
<td>Position at initial 10% peak amplitude</td>
</tr>
<tr>
<td>Pos_50_Beg</td>
<td>Position at initial 50% peak amplitude</td>
</tr>
<tr>
<td>Pos_90_Beg</td>
<td>Position at initial 90% peak amplitude</td>
</tr>
<tr>
<td>PosPeak</td>
<td>Position at peak amplitude</td>
</tr>
<tr>
<td>Pos_90_End</td>
<td>Position at final 90% peak amplitude</td>
</tr>
<tr>
<td>Pos_50_End</td>
<td>Position at final 50% peak amplitude</td>
</tr>
<tr>
<td>Pos_10_End</td>
<td>Position at final 10% peak amplitude</td>
</tr>
<tr>
<td>Pos_0_End</td>
<td>Position at final 0% peak amplitude</td>
</tr>
</tbody>
</table>

**Table 7.1:** Amplitude features.

### Energy

### Frequency
Table 7.2: Energy features.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMS_0_10_Beg</td>
<td>RMS between initial 0-10% peak amplitude</td>
</tr>
<tr>
<td>RMS_10_50_Beg</td>
<td>RMS between initial 10-50% peak amplitude</td>
</tr>
<tr>
<td>RMS_50_90_Beg</td>
<td>RMS between initial 50-90% peak amplitude</td>
</tr>
<tr>
<td>RMS_90_Peak_Beg</td>
<td>RMS between initial 90%:peak amplitude</td>
</tr>
<tr>
<td>RMS_Peak_90_End</td>
<td>RMS between final peak:90% amplitude</td>
</tr>
<tr>
<td>RMS_90_50_End</td>
<td>RMS between final 90-50% peak amplitude</td>
</tr>
<tr>
<td>RMS_50_10_End</td>
<td>RMS between final 50-10% peak amplitude</td>
</tr>
<tr>
<td>RMS_10_0_End</td>
<td>RMS between final 10-0% peak amplitude</td>
</tr>
</tbody>
</table>

Table 7.3: Frequency features.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DomFreq_0_Beg</td>
<td>Dominant Hz at initial 0% peak amplitude</td>
</tr>
<tr>
<td>DomFreq_10_Beg</td>
<td>Dominant Hz at initial 10% peak amplitude</td>
</tr>
<tr>
<td>DomFreq_50_Beg</td>
<td>Dominant Hz at initial 50% peak amplitude</td>
</tr>
<tr>
<td>DomFreq_90_Beg</td>
<td>Dominant Hz at initial 90% peak amplitude</td>
</tr>
<tr>
<td>DomFreq_Peak</td>
<td>Dominant Hz at peak amplitude</td>
</tr>
<tr>
<td>DomFreq_90_End</td>
<td>Dominant Hz at final 90% peak amplitude</td>
</tr>
<tr>
<td>DomFreq_50_End</td>
<td>Dominant Hz at final 50% peak amplitude</td>
</tr>
<tr>
<td>DomFreq_10_End</td>
<td>Dominant Hz at final 10% peak amplitude</td>
</tr>
<tr>
<td>DomFreq_0_End</td>
<td>Dominant Hz at final 0% peak amplitude</td>
</tr>
<tr>
<td>Q-3</td>
<td>Frequency tuning Q-3 dB at the pulse peak</td>
</tr>
<tr>
<td>Q-10</td>
<td>Frequency tuning Q-10 dB at the pulse peak</td>
</tr>
<tr>
<td>Ampl_H1</td>
<td>Amplitude of harmonic 1</td>
</tr>
<tr>
<td>Ampl_H2</td>
<td>Amplitude of harmonic 2</td>
</tr>
<tr>
<td>Ampl_H3</td>
<td>Amplitude of harmonic 3</td>
</tr>
</tbody>
</table>
Appendix C : Birds in this project

This appendix presents the different birds we have been using during this project, and shows the spectrographs of the songs or calls that have been used for our experiments.

Figure 7.1: DAB: Dusky Antbirds (*Cercomacra tyrannina*)
Figure 7.2: MAT: Mexican Antthrush (*Formicarius moniliger*)

Figure 7.3: BAS: Barred Antshrike (*Thamnophilus doliatus*)
Figure 7.4: DWAW: Dot-Winged Antwren (*Microphias quixensis*)

Figure 7.5: ACW: Acorn woodpeckers (*Melanerpes formicivorus*) (image (c) Mary Beth Stowe, reprinted with the permission of the author). The spectrogram of the *waka* call is shown.
C : Birds in this project
Appendix D : Field sites

Hastings reserve

This natural reservation located in Carmel Valley, near Monterey, north California. It belongs to the UC Berkeley. Calls of Acorn woodpeckers were recorded during the whole year. The ID of the calling bird was identified through color bands on both legs during recording. (source http://www.hastingsreserve.org )

Figure 7.6: Aerial map of the Hastings Reserve where the different acorn woodpecker territories that are used for this project.
James reserve

UC James Reserve, Idlywild, San Jacinto mountains. Electricity is generated using solar photovoltaics and generator. This reserve offers "On-line access to multiple micro-climate and reference weather stations, video feeds from cameras situated on towers and inside bird houses, fixed and mobile sensor platforms for soils, aquatic and above ground measurements of a variety of ecological processes." (source: http://www.jamesreserve.edu). This high-tech natural reservation, contains a network of robots and sensors developed by the Center for Embedded Networked Sensing monitor the forests vital signs, such as temperature, humidity and the biodiversity. The NIMS is robot that can move in three dimensions to collect data at different positions.

![Map of the James Reserve](image)

**Figure 7.7:** Map of the James Reserve with the different sensors that have been installed and the main NIMS transects.

Chajul, Monte Azules

This natural reservation located in the Monte Azules Biodiversity Reserve, Chiapas, Mexico, is a typical rainforest where the antbirds of interest in our project are located and were recorded. Good access to site, and facilities but still isolated from humans, so birds are not influenced too much by exterior influence.
Bibliography


