NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

AUTO DETECTION OF BLUE WHALE CALLS

by

Min-hsiu Wu

September 2023

Thesis Advisor:	Hong Zhou
Second Reader:	Paul Leary

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2023	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE
AUTO DETECTION OF BLUE WHALE CALLS

5. FUNDING NUMBERS

6. AUTHOR(S) Min-hsiu Wu

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

The thesis establishes an auto detection system that obtains Nonnegative Matrix Factorization (NMF) and harmonic structure detection techniques and tests it on the variety of recording data which is collected by the vector sensor in Monterey Bay. This system is currently examined specifically on blue whale B calls. In the thesis, we find that a NMF can be implemented in different update rules and initialization methods and yield different output which depends on input data. In addition, we apply Genetic Algorithm (GA) to reduce the error which is produced by NMF. Last, Harmonic Product Spectrum (HPS) takes place to examine whether the harmonic structure of blue whale B calls exists in the output of NMF and GA.

In order to reduce human resource on physically checking the existence of blue whale, the goal is to find the best combination of algorithms that identifies the calls with lowest rate of false alarm. In further research, the performance can be improved by applying parallel computation. And manipulating the data of vector sensor to locate the source of sound can increase the accuracy of this detection system. Moreover, obtaining the detection of various marine mammals is needed.

14. SUBJECT TERMS genetic algorithm, nonnegative matrix factorization, harmonic product spectrum	**15. NUMBER OF PAGES** 81
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

AUTO DETECTION OF BLUE WHALE CALLS

Min-hsiu Wu
Lieutenant Junior Grade, Taiwan Navy
BSM, National Defense University, 2018

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN APPLIED MATHEMATICS

from the

NAVAL POSTGRADUATE SCHOOL
September 2023

Approved by: Hong Zhou
Advisor

Paul Leary
Second Reader

Francis X. Giraldo
Chair, Department of Applied Mathematics

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The thesis establishes an auto detection system that obtains Nonnegative Matrix Factorization (NMF) and harmonic structure detection techniques and tests it on the variety of recording data which is collected by the vector sensor in Monterey Bay. This system is currently examined specifically on blue whale B calls. In the thesis, we find that a NMF can be implemented in different update rules and initialization methods and yield different output which depends on input data. In addition, we apply Genetic Algorithm (GA) to reduce the error which is produced by NMF. Last, Harmonic Product Spectrum (HPS) takes place to examine whether the harmonic structure of blue whale B calls exists in the output of NMF and GA.

In order to reduce human resource on physically checking the existence of blue whale, the goal is to find the best combination of algorithms that identifies the calls with lowest rate of false alarm. In further research, the performance can be improved by applying parallel computation. And manipulating the data of vector sensor to locate the source of sound can increase the accuracy of this detection system. Moreover, obtaining the detection of various marine mammals is needed.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

AG	Array Gain
ALS	Alternating Least Squares
ANLS	Alternating Non-negative Least Squares
ASW	Anti-Submarine Warfare
CI	Call Index
CNN	Convolutional Neural Network
CREPE	Convolutional Representation for Pitch Estimation
DFT	Discrete Fourier Transform
DI	Directivity Index
DT	Detection Threshold
GA	Genetic Algorithm
HPC	High-Performance Computer
HPS	Harmonic Product Spectrum
IMGA	Island Model Genetic Algorithm
MARS	Monterey Accelerated Research System
MBARI	Monterey Bay Aquarium Research Institute
MMPA	Marine Mammal Protection Act
MU	Multiplicative Update
NL	Noise Level

NMF Non-negative Matrix Factorization

NMFS National Marine Fisheries Service

NOAA National Oceanic and Atmospheric Administration

NNSVD-LRC Non-negative SVD with Low-Rank Correction

NPS Naval Postgraduate School

PAM Passive Acoustic Monitoring

PC-NMF Periodicity Coded Non-negative Matrix Factorization

SL Source Level

STFT Short-Time Fourier Transform

SUSWO Stanton Undersea Shallow Water Observatory

SVD Singular Value Decomposition

TFCNN Temporal-Frequency Attention Based Convolutional Neural Network Model

TL Transmission Loss

YAAPT Yet Another Algorithm for Pitch Tracking

Acknowledgments

I really appreciate the help from my advisor, Professor Hong Zhou. The original concept of this thesis was really broad. You helped me to narrow down the problem and gave me lots of advice when I was struggling in the choice of algorithms. Furthermore, I want to thank my second reader, Dr. Paul Leary. The data set you provided is also the key for this thesis.

Next, I am grateful to the Taiwan Navy which gave me this opportunity to study abroad. Also, I extend my appreciation to everyone in the Naval Meteorological and Oceanographic Office (NMOO). Everything I learned from you guys is the foundation of this study. During these two years, I appreciate the colleagues who are from Taiwan and also study here. I will always remember the time we were together.

Lastly, I am thankful to my family. You all always support what I really want and point out the direction of my life. I also feel sorry that I was not able to take care of you during the time I studied here. I will spend more time with you all when I get back.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

To protect marine mammals from some harmful human activities, it is important to recognize the animal's existence in the area before executing such activities. For instance, an activity related to active sonar systems can possibly stress marine mammals within certain distance. With that in mind, distance and sound intensity should be considered before implementing these systems. Under the Marine Mammal Protection Act (MMPA), enforced mainly by National Oceanic and Atmospheric Administration (NOAA) Fisheries, also known as the National Marine Fisheries Service (NMFS), regulates human activities that could potentially or directly harm marine mammals in the waters of the United States [1]. Thus, work monitoring such a large designated area over a long period and understanding the behaviors of marine mammals can undoubtedly require considerable manpower.

Tracking the activities and presence of marine mammals can be implemented in several ways. One is by reports from people who observe them. However, this method could give inaccurate information if the report is not offered by those who are not experts in marine biology. Meanwhile, the likelihood of running into these creatures can be extremely small. Consequently, an alternative way is to deploy hydrophones in several areas, and analyze the recorded data over time. This method is also called Passive Acoustic Monitoring (PAM). Without being physically present at a location, people can track the mammals in this way. Furthermore, an unmanned system with automatic sound detection can efficiently reduce the human power needed to examine those data.

In this research, we establish and test a system that can automatically detect marine mammals by acoustic data. To narrow this research, we apply the developed system to track blue whales by a specific type of call. Thus, in the following sections, we talk about the harmonic structure of blue whale calls and methods that have been used in the acoustics recognition field.

1.1 Background Knowledge on Blue Whale Calls

A blue whale can create a moan at 12.5 Hz to 200 Hz that lasts up to 36 seconds according to [2]. There are three types of blue whale calls that repeatedly occur over the low frequency range from 10 Hz to 100 Hz [3]. Figure 1.1 illustrates the frequency of the calls. Notably, *B* calls conform to a harmonic structure that can help a manipulator or an automatic detection system verify the existence of blue whales in an area of interest. So, we consider blue whale *B* calls in particular as a good approach that makes visualization and review of the system result convenient. The data set that is assessed in our research is collected by Monterey Accelerated Research System (MARS) placed on the ocean floor of Monterey Bay area in California. Given the location of the system, abundant acoustic data is available to use, as discussed in Chapter 2.

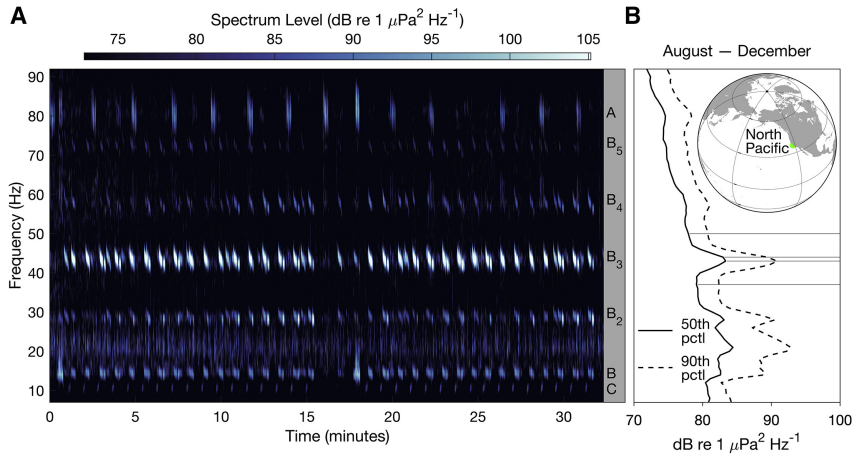


Figure 1.1. Blue Whale Calls spectrograms in the low frequency range. Source: [3]. This data is being collected in Monterey Bay area and shows the frequency signatures of blue whale calls. Blue whale calls in the frequency range under 100 Hz consist of A, B, and C calls, whereas B call is a harmonic structure. The fundamental frequency B is around 15 Hz, 30 Hz for B_2, 45 Hz for B_3, 60 Hz for B_4, and 75 Hz for B_5. Furthermore, B_3 has the strongest intensity over all signature frequencies of blue whale calls in certain frequency range, which forms the focus of our research with the specific algorithms.

1.2 Literature Review

PAM was originally applied to Anti-Submarine Warfare (ASW) after World War II. In addition to tracking the activities of undersea targets, it is now being utilized for the analysis of the characteristics of various types of artificial or natural noise [4]. Over time, researchers have proposed numerous state-of-art algorithms to make this process more efficient and accurate. In [5] and [6], both researches include the concept of Call Index (CI) to track the presence of blue whales in Monterey Bay, California, and understand the relation between the occurrence and environmental factors with the same data set as we use. The idea of CI is to calculate the ratio of maximum acoustic intensity to the average intensity without the call within respective frequency range [6]. The presence of blue whales is determined according to the value of CI.

As the machine learning became a mainstream technology in the past decade, people started focusing on its applications to PAM. One of these approaches is to apply an algorithm that classifies the source without any prior knowledge of that sound source, even though this method cannot identify what kind of source produced the sound. The idea is to filter out all sources in the data and provide a clear picture for a researcher to produce a more accurate estimation. In [7], they introduce the conventional classifiers such as nearest-neighbor methods with different distance (divergence) functions and apply the classifiers to a data set with several types of bird calls to see how accurate these methods are at distinguishing different calls. In [8], Temporal-Frequency Attention Based Convolutional Neural Network Model (TFCNN) is presented to classify environmental sound by referencing time and frequency frames. The TFCNN approach yields the characteristics of different sources and allows the researchers to classify them. Although this method is inaccurate on intermittent sounds, it shows great predictive ability on continuous ones. To emphasize the temporal feature sounds, [8] suggests the time attention mechanism is required in future work.

Besides the classifier and neural network model, another way to scrutinize a spectrogram is by matrix factorization. Specifically, [9] introduces Periodicity Coded Non-negative Matrix Factorization (PC-NMF) to understand biodiversity by recording sound. PC-NMF is an algorithm that factorizes an input spectrogram into two matrices and applies Discrete Fourier Transform (DFT) to one of the matrices. An example shown in Figure 1.2 illustrates the concept of PC-NMF. The detail of factorization part called Non-negative Matrix Factor-

ization (NMF) is explained in Chapter 3, which includes the initialization, different update rules, and error reduction process. The main idea of NMF is to decompose the spectrogram into matrices that individually contain information on frequency and time. Each column and corresponding row spaces in these two matrices are considered as representations of sources or noise. In order to separate the signal from the noise, DFT is used on the row space of the time matrix to obtain the temporal information. As a result, researchers can differentiate the source from the noise by checking the result of PC-NMF.

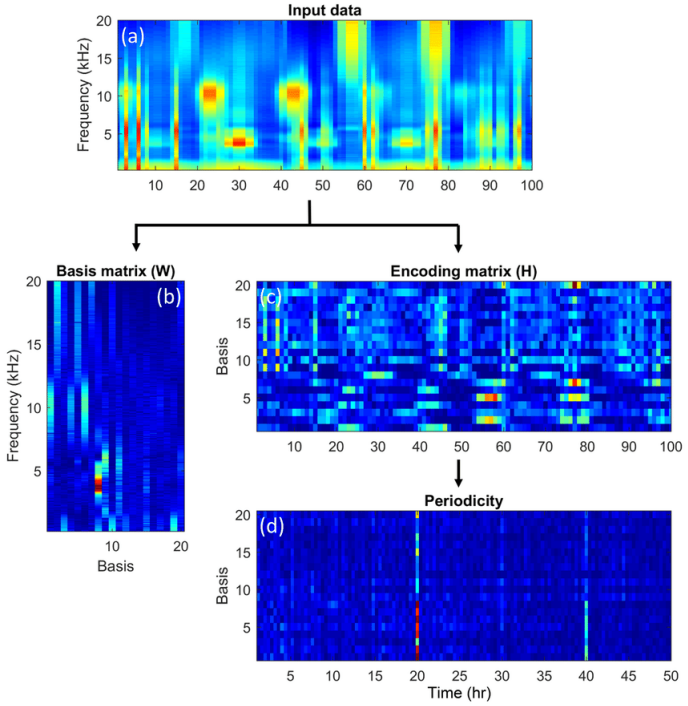


Figure 1.2. An example of PC-NMF. Source: [9]. (a) Input spectrogram; (b) matrix with frequency information in column space; (c) matrix with time information in row space; (d) periodicity matrix. By NMF, the input matrix is factorized in to matrices with details on frequency and time. With the aim of identifying the source and noise, the periodicity matrix is given by applying DFT to the matrix with time information.

From the NMF result, frequency and time matrices are obtained. PC-NMF utilizes the time information to understand the input sound. Different from [9], we focus on the harmonic structure of blue whale *B* calls by the frequency matrix. The design of this system in our

research includes a pitch detection algorithm that helps the system recognize any harmonic structure in the column space of the W matrix shown in Figure 1.2. Here, pitch also called fundamental frequency is a term that indicates the lowest frequency in the harmonic structure. In [10]'s work, they present two conventional algorithms, such as probabilistic YIN Yet Another Algorithm for Pitch Tracking (YAAPT), and Convolutional Representation for Pitch Estimation (CREPE) which is capable of pitch detection with the Convolutional Neural Network (CNN). All of these algorithms yield great results without redundant signal processing. However, in our system, the input spectrogram has been separated into frequency and time matrices, and we also need an algorithm that takes less computational cost. Consequently, we include the Harmonic Product Spectrum (HPS), which is mentioned in [11], to look over the frequency matrix, W, and check for the existence of blue whale B calls. The details about the HPS are explained in Chapter 3.

In this thesis, we analyze the data that includes blue whale calls by NMF with different update rules. After that, HPS checks each column space of the W matrix produced by NMF to see whether the blue whale calls occur in this time frame. Finally, the result of this system returns a low probability of false alarm. That means this system gives an alarm if the recorded sound is truly a blue whale call.

1.3 Organization

This thesis is organized into four additional chapters and one appendix. Chapter 2 provides details about data set and briefly introduces the difference between hydrophone and vector sensor, as well as some basic knowledge of underwater acoustics. In Chapter 3, we include the NMF, Genetic Algorithm (GA), and HPS, which factorize input spectrogram into two matrices, decrease the error between the original and estimated spectrograms, and seek out the harmonic structure of blue whale B calls inside the frequency matrix, respectively. The method of data processing and performance are presented in Chapter 4. Chapter 5 concludes this research and discusses areas for future work. It is worth noting that the developed system is programmed in Matlab, and the source code for the system is provided in the appendix.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Data Acquisition

The Monterey Bay Aquarium Research Institute (MBARI), Stanton Undersea Shallow Water Observatory (SUSWO), and Naval Postgraduate School (NPS) collected the data used in this study during the past decades. This acoustic data, which was subject to the High-Performance Computer (HPC), Hamming, on the NPS campus, was compiled into an SQLite database. A single Geospectrum M20-105 vector sensor located outside Monterey Bay at 1,000 meters water depth was used to record and transmit the data to the MARS, which is run by MBARI [12], [13]. This is where the presence of marine mammals are likely to be present due to the upwelling current that goes along Monterey Canyon from west to east, bringing nutrients from the bottom to the surface, helping the growth of phytoplankton, and establishing the food chain for this area. Figure 2.1 suggests where the sensor is and how it is cabled to the system.

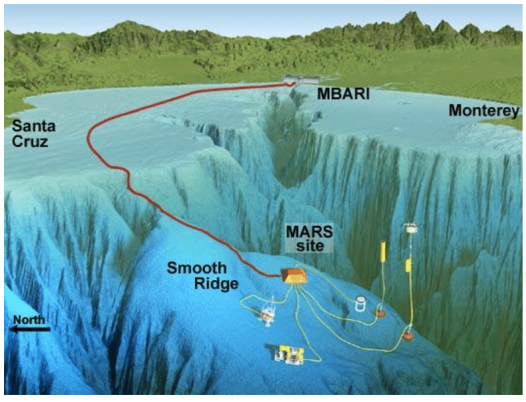


Figure 2.1. View of MARS system in Monterey Bay, California. Source: [13]. Along the west coast of United States of America, the current moves southward with cold water from the Arctic area. By wind stress curl, the upwelling current along the coast is formed and brings the nutrients that create the food chain and ecological system [14]. In the middle of Monterey Bay, the Monterey Canyon allows the upwelling current to penetrate the bay area. The MARS system is located where the upwelling current and most marine creatures, including marine mammals, pass.

2.1 Hydrophone

Before delving into a discussion of the vector sensor, we would like to introduce the hydrophone because it is one part of the vector sensor. The hydrophone is one of the most widely used underwater receivers, and the way it measures the sound wave in water is by using piezoelectric material, which creates small electric signals when pressure on the material changes. Hence, a pre-amplifier is required and connected to the hydrophone for greater sensitivity. Generally, there are two types of hydrophone, one is omnidirectional, and the other is directional [15]. Omnidirectional hydrophone is able to receive signals from almost all directions as shown in Figure 2.2, while the directional one is highly concentrated on where it faces.

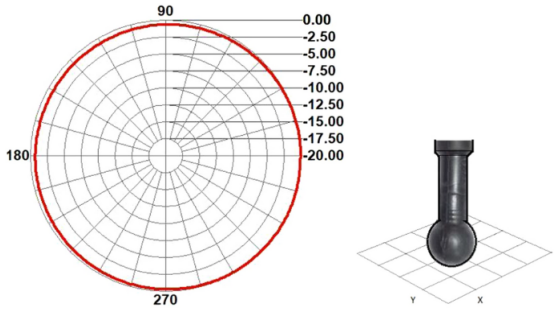


Figure 2.2. An example of directivity of omnidirectional hydrophone. Source: [16]. The hydrophone on the right is being tested on a specific frequency. The result is as shown on the left. Those numbers affixed to the circles with different diameters stand for the intensities that the hydrophone detects. According to the result plotted in red, the omnidirectional hydrophone can detect the sound wave from all directions.

Despite all the previously advantages of the hydrophone, a single omnidirectional hydrophone cannot determine the direction of a source since the reception is from all direction. By contrast, the directional one does not detect the sound if the sound wave is not aimed directly toward it. To fix this problem, one might deploy an array of omnidirectional hydrophones. With the array, the direction can be approximated by calculating the difference of the arrival time and the relative position between the hydrophones [17]. Another way is to use a vector sensor, which is to be discussed in the following section.

2.2 Vector Sensor

Vector sensors measure not only pressure change (sound waves) but also particle motion. Typically in addition to one pressure sensor, at least two particle motion sensors are embedded in a vector sensor. That is, we can view a vector sensor as carrying at least two hydrophones (the pressure sensor is not included since it measures a different quantity). With this feature, a single vector sensor can derive the bearing of sources, while a hydrophone may or may not be able to yield precise orientations.

As shown in Figure 2.3, the Geospectrum M20-105 vector sensor is installed in MARS. According to the specification sheet [18], the sensor has three dipole sensors that detect particle motion and an omnidirectional hydrophone that measures the sound wave. The frequency response range is from 1 Hz to 3,000 Hz. However, the official site suggests that frequencies below 10 Hz may be noise, which is not reliable.

With the vector sensor, we can track the activities of blue whales by listening to their calls, but things can be uncertain when those targets are distant from the receiver.

Figure 2.3. Geospectrum M20-105 Vector Sensor. Source: [18]. A single vector sensor can measure the pressure and direction of a sound wave using pressure sensor and particle motion sensors. The Geospectrum M20-105 vector sensor can detect a frequency range of 1 Hz to 3,000 Hz. Thus, it is capable of detecting marine mammals, particularly blue whales as the frequency of their calls is between 10 Hz to 100 Hz.

2.3 Underwater Acoustics

The sound of blue whale call being received might travel several miles and experience multiple instances of bottom and surface reflections. Typical frequencies associated with underwater acoustics vary from 10 Hz to 1 MHz. When sound frequencies are below 10 Hz, they penetrate deep into the seabed. When sound frequencies are above 1 MHz, they are absorbed very quickly. Thus, the propagation of sound is affected by many factors. First, the speed of sound in the sea can be affected by temperature, salinity, and pressure. The equation to describe the velocity of sound, is itself complicated. For example, in the

following equation 2.1,

$$c = 1449.2 + 4.6T - 0.055T^2 + 0.00029T^3 + (1.34 - 0.01T)(S - 35) + 0.016z \quad (2.1)$$

where several terms are being shown, c is the sound speed in meters per second (m/s), temperature T in $°C$, salinity S in ‰, and depth z in meters [19]. In addition, equation 2.1 is only applicable when the temperature is between $1°C$ and 35 $°C$, salinity between 0 ‰ and 45 ‰, and depth between 0 meter and 1000 meters. However, those quantities in equation 2.1 are not constant horizontally or vertically. Therefore, refraction gets involved along the propagation according to Snell's Law [20],

$$c_{z_0} cos\theta_z = c_z cos\theta_{z_0} \quad (2.2)$$

where z_0 and z stand for the present depth and the depth in following step, respectively, sound speed c_z at depth z with equation 2.1, and grazing angle θ_z with surface.

Next, before reaching the receiver, the sound wave decreases in intensity because of the diffraction from the bottom, surface, or bubbles in the water, as well as transmission loss. Last, the detection threshold of the receiver, and the noise level of environment are the other elements that decide whether the receiver can detect the sound of blue whale calls. Overall, we can get passive-sonar equation by following [21]:

$$SL - TL = NL - DI + DT \quad (2.3)$$

where SL represents the Source Level (SL), TL is the Transmission Loss (TL), NL is the Noise Level (NL), DI is the receiving Directivity Index (DI), and DT is the Detection Threshold (DT). All quantities in equation 2.3 are measured in decibels (dB), which is defined as

$$dB = 10 \log_{10} \frac{I}{I_0} = 20 \log_{10} \frac{p}{p_0} \quad (2.4)$$

where I_0 is the reference intensity, I is the intensity being estimated, p_0 is reference pressure, and p is the estimated pressure. For an underwater plane wave, p_0 is typically 10^{-6} pascals (Pa) or a micropascal (μPa) according to [20]. In our research, the sound pressure level of blue whale calls (moans) is on average 188 dB for SL [2]. TL is calculated according to the impedance and pressure of the source and the receiver. The details can be found in [20].

TL is dependent on the distance between the source and the receiver. Specifically, longer distance is related to larger TL. DI is associated with the number of hydrophones in an array. With certain distance between each hydrophones, the intensity or pressure level of some frequencies will be enhanced, which is called Array Gain (AG). In general, DI is computed in the following way [20]:

$$DI = 10\log_{10}(n) \tag{2.5}$$

where n is the number of hydrophones. Since we obtain data from only one vector sensor in this study, this yields $DI = 0$. The next factor is DT which is associated with probability theory and related to NL. Briefly, DT is expressed in [20]:

$$DT \equiv \mathrm{SNR}_{DT} = 10\log_{10}\frac{S}{N_0} \tag{2.6}$$

where S is the intensity of the signal that the receiver gets, and N_0 is the noise level. Both are estimated within the 1-Hz bandwidth. To get DT, we need to understand the environmental situation around the receiver and get N_0. Since the noise level is varies throughout time, the probability density function of noise level is needed. After this process, we can obtain DT statistically.

To sum up, the receiver is able to detect blue whale calls when $SL + DI$ exceeds the sum of all the rest of the terms in equation 2.3. However, detection could be difficult since traffic noise and other natural noise can dominate the blue whales' calls in certain frequency ranges as shown in Figure 2.4. Therefore, detection can be challenging and dependent on several factors most of the time.

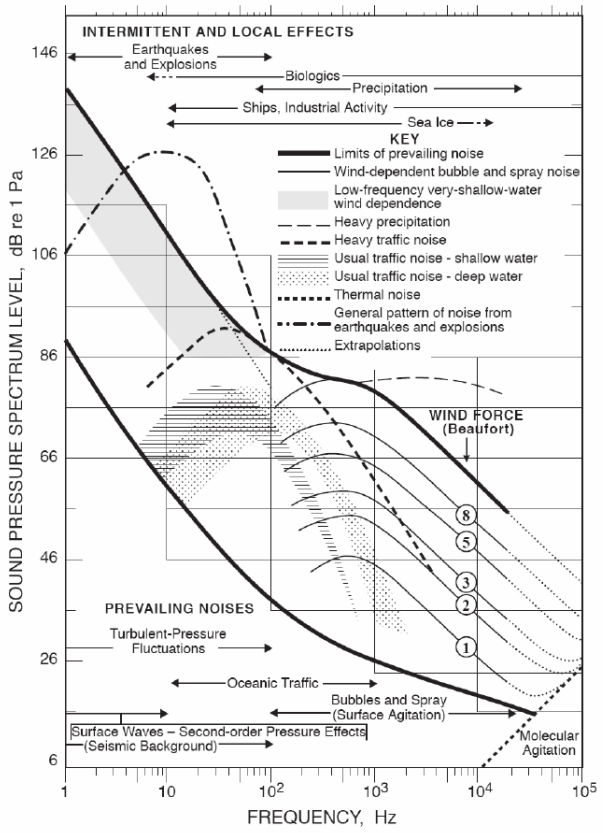


Figure 2.4. Composite of ocean ambient noise. Source: [17]. Based on [2], [3], the frequency range of blue whale B calls is around 10 Hz to 100 Hz, which overlaps with the range of marine traffic. This could make the detection of blue whale calls more difficult when the noise level of traffic dominates this frequency range and makes DT higher.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Methodologies

In this chapter, we introduce the methods that we used to auto detect blue whale calls. First, using NMF, an input spectrogram is decomposed into two matrices. One of these two matrices carries the frequency information of the spectrogram while the other holds the time information. With these matrices, we can determine whether this data contains the whale calls. Next, we apply GA to filter out the columns and corresponding rows of matrices since there exists error between the product of matrices and the input spectrogram. Finally, HPS is applied to determine whether blue whale calls occur or not in the data.

3.1 Non-negative Matrix Factorization

The Non-negative Matrix Factorization (NMF) algorithm is widely used in signal and image processing. It is a low rank approximation algorithm to estimate a data matrix. By decomposing a data matrix, researchers can see the properties of the input data.

Consider that we have a matrix $X_{n\times m}$ with n rows and m columns in which all elements are non-negative. Find the non-negative matrices $W_{n\times k}$ and $H_{k\times m}$ such that

$$X_{n\times m} \approx W_{n\times k} H_{k\times m} \tag{3.1}$$

The algorithm iteratively updates W and H matrices subject to

$$\min_{W\geq 0, H\geq 0} \|X - WH\|_F^2 \tag{3.2}$$

where the subscript, F, stands for the Frobenius norm as shown in equation 3.3, and all elements in the W and H matrices are all greater than or equal to zero.

$$\|X - WH\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m} |(X - WH)_{ij}|^2} \tag{3.3}$$

Previous research by Lee and Seung [22] indicates that the approximation under Multiplicative Update (MU) is non-increasing. Since we use the squared- Frobenius norm as the cost function, we can apply the following rule [22]

$$H_{\alpha j} \leftarrow H_{\alpha j} \frac{(W^T X)_{\alpha j}}{(W^T W H + \epsilon)_{\alpha j}} \qquad W_{i\alpha} \leftarrow W_{i\alpha} \frac{(XH)_{i\alpha}}{(WHH^T + \epsilon)_{i\alpha}} \qquad \forall \alpha \leq k, i \leq n, j \leq m \tag{3.4}$$

to update the elements in W and H, where ϵ is a small number to prevent zero division.

The non-increasing characteristic is proved by Lee and Seung, but it has been pointed out that the convergence of MU may be unstable [23]. We discuss this issue later.

3.1.1 Improved SVD-based Initialization

Lee and Seung [22] have proven the convergence of NMF, but the efficiency of NMF is also important for us. Particularly, we need to take the number of iterations of NMF into account. This is associated with the method of initialization, multiplicative rule, and cost function. In this paper, we focus on the initialization of NMF. Ordinary NMF has the non-negative matrices, $W_{n\times k}$ and $H_{k\times m}$, being randomly produced. With this random process, we could not be sure whether the algorithm ends up with an efficient local minimum. To solve this problem, many initialization algorithms were introduced. In general, we can classify these algorithms into four categories: random, heuristic, clustering, and low-rank schemes [24]. In addition, k is the parameter that is not determined. We do not know how many characteristics are in the input matrix, we only know that $k < \min(n, m)$ according to [25].

Given the problems mentioned, we decided to consult the work by Atif, Qazi and Gillis [26]. Different from a regular Singular Value Decomposition (SVD) based method, they presented the Non-negative SVD with Low-Rank Correction (NNSVD-LRC) NMF. NNSVD-LRC uses truncated SVD, which yields the best rank-k approximation of the input matrix [27], as a reference to initialize the W and H matrices. In this case, NMF can converge to local minimum faster than random initialization does. To accomplish this, it works as follows:

Let r be the rank of input matrix $X_{n\times m}$ in equation 3.1.

1. Apply truncated SVD approximates $X_{n\times m}$ by

$$X_{n\times m} \approx U_p \Sigma_p V_p^T \tag{3.5}$$

where $p = \lfloor \frac{r}{2} + 1 \rfloor$ ($\lfloor \cdot \rfloor$ is the floor division, which gives the greatest integer that is less than or equal to the result of operation inside); $U_p \in \mathbb{R}^{n\times p}$; $V_p \in \mathbb{R}^{p\times m}$; and $\Sigma_p \in \mathbb{R}^{p\times p}$ is a diagonal matrix with singular values on its diagonal. This is not an ordinary SVD operation as the Σ matrix does not hold the size same as X or X_p.

2. Now, let W and H be n-by-r and r-by-m matrices, respectively. To get the initialized W and H, the next step is to reduce the three terms in equation 3.5 to two terms. Therefore, [26] introduces Y_p and Z_p, where

$$Y_p = U_p \Sigma_p^{1/2} \qquad Z_p = \Sigma_p^{1/2} V_p^T \tag{3.6}$$

3. Assign absolute values from the first column of Y and row of Z to W and H matrices.

$$W_{i1} = |Y_{i1}|, \quad H_{1j} = |Z_{1j}| \qquad \forall i, j \in \mathbb{N} \tag{3.7}$$

where $i \leq n$ and $j \leq m$.

4. Use the following rule to fill in elements of the W and H with the remaining ranks of $Y_p$and Z_p:

$$\begin{cases} W_{i\alpha} = Y_{i\beta}^{(\geq 0)}, \quad H_{\alpha j} = Z_{\beta j}^{(\geq 0)} & \text{if } \alpha \text{ is even} \\ W_{i\alpha} = |Y_{i\beta}^{(\leq 0)}|, \quad H_{\alpha j} = |Z_{\beta j}^{(\leq 0)}| & \text{if } \alpha \text{ is odd} \end{cases}, \quad where \quad \beta = \lfloor \frac{\alpha}{2} + 1 \rfloor \qquad \forall 2 \leq \alpha \leq r \tag{3.8}$$

With NNSVD-LRC, the objective function is now changed to

$$\min_{W\geq 0,, H\geq 0} \|X_p - WH\|_F^2 \tag{3.9}$$

where $X_p = Y_p Z_p$.

3.1.2 Alternating Least Squares Algorithm

M.W. Berry et al. [23] mentions three types of update rules, MU, Gradient Descent, and Alternating Least Squares (ALS) algorithms, for NMF. D. Lee and H.S. Seung [22] introduces MU rule, which is given in equation 3.4. However, [28], [29] show that the MU rule takes more iterations than ALS and Gradient Descent to converge. Furthermore, [23] suggests that the convergence of the MU rule may or may not meet the local minimum. Therefore, we adopt ALS as another update method and see whether these two rules converge to the local minimum and how we can improve the result to help us detect blue whale calls.

The ALS algorithm is a useful tool for matrix factorization. It is specifically called Alternating Non-negative Least Squares (ANLS) in NMF in order to keep all entries in W and H greater than or equal to zero [30]. Basic ALS is implemented as follows [23]:

1. Randomly initialize or apply NNSVD-LRC to the W matrix.
2. Fix W and update H by solving equation 3.10.

$$W^T W H = W^T X \tag{3.10}$$

3. Set all elements in H to be non-negative and apply the objective function, which is given in equation 3.2.
4. Fix H and update W by solving equation 3.11.

$$H H^T W^T = H X^T \tag{3.11}$$

5. Apply the objective function as step 3.
6. Repeat steps 2 to 5 until the stop criterion is met.

The stop criterion determines how many iterations the NMF process makes. Typically, NMF is stopped when the difference between the current value of the objective function and the previous iteration's value is less than a small number, ϵ, which depends on scenario.

3.1.3 An Example of NNSVD-LRC

A simple demonstration of NNSVD-LRC is as follows:

Let A be a 3×2 input matrix with $rank = 2$, and find the initialized $W_{3\times2}$ and $H_{2\times2}$ by NNSVD-LRC, where

$$A_{3\times2} = \begin{bmatrix} 5 & 1 \\ 4 & 2 \\ 8 & 1 \end{bmatrix}$$

First, apply the truncated SVD with $p = \lfloor \frac{2}{2} + 1 \rfloor = 2$.

$$A_{3\times2} \approx U_{3\times2}\Sigma_{2\times2}V^T_{2\times2} = \begin{bmatrix} -0.4878 & 0.0125 \\ -0.4131 & -0.8842 \\ -0.769 & 0.467 \end{bmatrix} \begin{bmatrix} 10.4532 & 0 \\ 0 & 1.3152 \end{bmatrix} \begin{bmatrix} -0.9799 & -0.1993 \\ 0.1993 & -0.9799 \end{bmatrix}$$

Then, obtain $Y_{3\times2}$ and $Z_{2\times2}$ by

$$Y_{3\times2} = U\Sigma^{1/2} = \begin{bmatrix} -0.4878 & 0.0125 \\ -0.4131 & -0.8842 \\ -0.769 & 0.467 \end{bmatrix} \begin{bmatrix} 3.2331 & 0 \\ 0 & 1.1468 \end{bmatrix} = \begin{bmatrix} -1.5771 & 0.0143 \\ -1.3356 & -1.0139 \\ -2.4864 & 0.5356 \end{bmatrix}$$

$$Z_{2\times2} = \Sigma^{1/2}V^T = \begin{bmatrix} 3.2331 & 0 \\ 0 & 1.1468 \end{bmatrix} \begin{bmatrix} -0.9799 & -0.1993 \\ 0.1993 & -0.9799 \end{bmatrix} = \begin{bmatrix} -3.1683 & -0.6443 \\ 0.2285 & -1.1238 \end{bmatrix}$$

With $Y_{3\times2}$ and $Z_{2\times2}$, initialize W and H according to equations 3.7 and 3.8.

$$W = \begin{bmatrix} 1.5771 & 0.0143 \\ 1.3356 & 0 \\ 2.4864 & 0.5356 \end{bmatrix}, \quad H = \begin{bmatrix} 3.1683 & 0.6443 \\ 0.2285 & 0 \end{bmatrix}$$

These are the initialized W and H. Now, according to equation 3.9, the objective function has become

$$\min_{W\geq0,,H\geq0} \|A_p - WH\|_F^2$$

$$\text{where } A_p = YZ = \begin{bmatrix} -1.5771 & 0.0143 \\ -1.3356 & -1.0139 \\ -2.4864 & 0.5356 \end{bmatrix} \begin{bmatrix} -3.1683 & -0.6443 \\ 0.2285 & -1.1238 \end{bmatrix} = \begin{bmatrix} 5.000 & 1.000 \\ 4.000 & 2.000 \\ 8.000 & 1.000 \end{bmatrix}.$$

Notice that A_p looks the same as A since this is a simple case.

■

Eventually, NNSVD-LRC NMF yields W and H with a certain k, which is $rank(X)$. In the meantime, W contains frequency information in the column space, and H holds time information in the row space, respectively. The advantage of using NNSVD-LRC as the initialization algorithm is that it helps researchers decide the value of k, which is the dimension of the column space of W or row space of H. Nevertheless, we can still find an error between the input spectrogram, X, and WH since the algorithm is a low-rank approximation. Thus, the following section present the algorithm that lowers the error by reducing the size of W and H.

3.2 Genetic Algorithm

The Genetic Algorithm (GA) is an evolutionary algorithm used to solve combinatorial optimization problems. Examples of these kinds of problems include the traveling salesman problem, vehicle routing problem, and other problems without algorithmic solvers. The idea of GA is similar to that of natural selection. Individuals evolve with each generations while only the ones who adapt to the environment the best survive. GA inherits the feature of natural selection by beginning with an initial population, which is a set of individuals. Here, we let the number of individuals in a population be $\lfloor \frac{r}{2} \rfloor + 1$,where r is the rank of the input matrix, X. Each individual containing a chromosome is considered as one of the solutions of GA. The whole endeavor is to find the individual who fits or survives under specific conditions the best. However, it is not guaranteed that this individual is the global minimum for the problem since GA is established upon stochastic models. As a result, parallel computation is one solution that processes GA in different cores, treats each core as an island which holds its own population, and interchanges the best individual between the cores. The diversity of individuals helps GA get closer to an optimal solution. This

algorithm is generally called the Island Model Genetic Algorithm (IMGA) [31]. Here, we only concentrate on ordinary GA since this research is still in the early stage.

GA consists of four steps [32]:

1. *Selection*: From the population, randomly choose pairs of individuals to reproduce the next generation.
2. *Reproduction*: Pairs of individuals selected by previous step recombine (crossover) and mutate each other's chromosomes.
3. *Evaluation*: Evaluate the individuals by fitness function.
4. *Replacement*: Kill the individuals of the older generation and have the new generation process these four steps until the best individual meets the threshold.

The cycle of GA is shown in Figure 3.1.

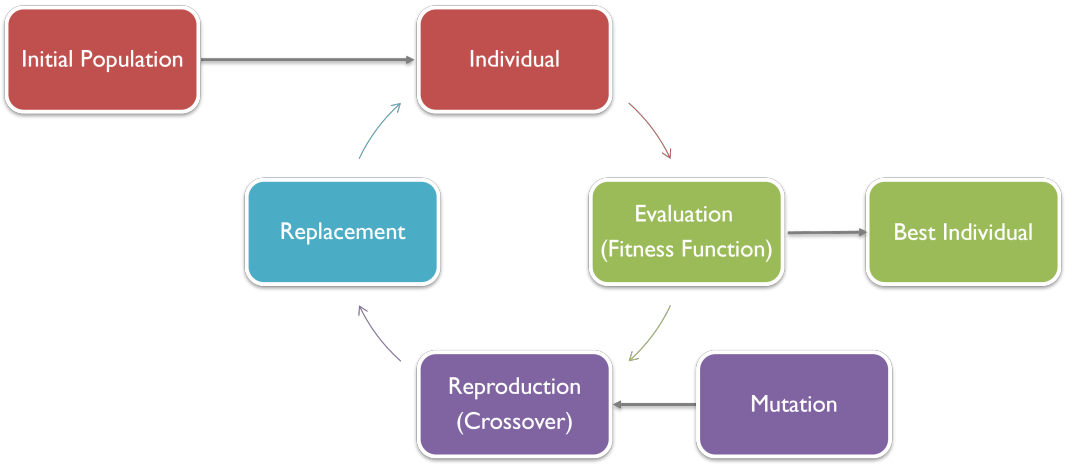


Figure 3.1. The simplified GA cycle. Adapted from: [32]. First, GA randomly produces the initial population as input. Each individuals is evaluated by the fitness function, and the best one of this generation will be replaced if there is a better one in the following generation. Next, two arbitrary individuals in the population become a pair, crossover their chromosomes, and yield the next generation. Meanwhile, during reproduction, there is a chance that a mutation happens in each pair. The probability of mutation depends on the scenario and is relevant to how many generations GA will take to reach the threshold. Last, the population is replaced by a new generation of children, and GA starts another new cycle until the best individual shows up.

3.2.1 Individuals

As mentioned, each individuals is a solution in GA. Common solutions between different individuals are allowed, but this can cause uncertainty if their number is too large. In the other words, the diversity of these individuals is too small to yield a better result. Individuals are randomly generated. An individual contains multiple factors that are isomorphic to a sequence of genes on a chromosome. There are various types of chromosomes, depending on how researchers encode the information. In this paper, we liken individuals to bit strings. The length of each bit string is r, which is the rank of input matrix, V. Meanwhile, we change the entries of W and H by the following:

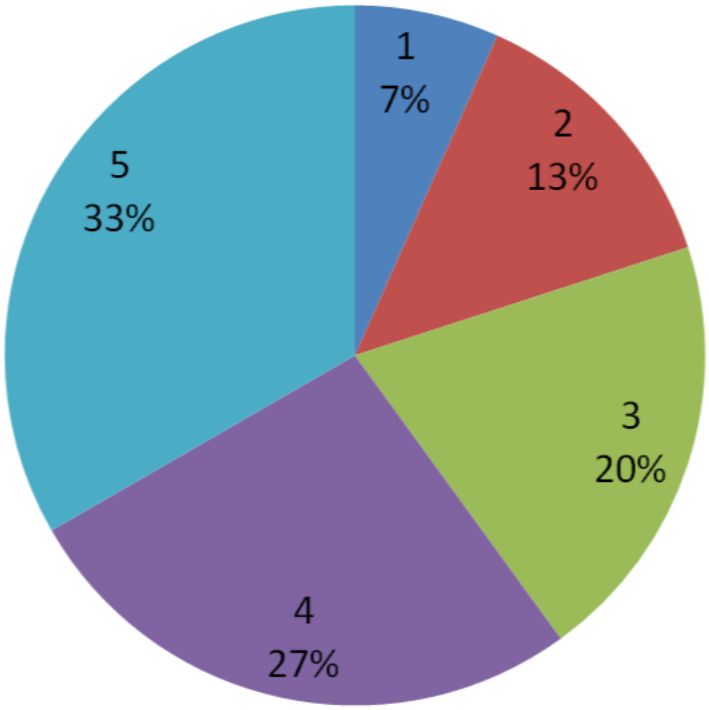
Let B be an individual, which is represented by a bit string with the length of r. And let

b_k be the element of B, where $1 \leq k \leq r$.

$$\begin{cases} W_{ik} = 0, & H_{kj} = 0 & \text{if } b_k = 0 \\ W_{ik} = W_{ik}, & H_{kj} = H_{kj} & \text{if } b_k = 1 \end{cases} \tag{3.12}$$

As to enable individuals to evolve with better results, individuals are divided into pairs (selection), and within each pair there are crossover chromosomes. In practice, random selection could take too many generations to get the best individual. It is therefore better to apply some rules to weigh individuals according to their performance which is evaluated by the fitness function. That is to say, the individuals with higher scores are much more likely to be selected.

Depending on the problem, the methods of crossover can be single point, double point, or others that algorithmically or heuristically choose the points at which to exchange portions of chromosomes. The other factor that plays an important role is mutation. Mutation takes place within chromosomes according to which probability distribution a researcher applies. This determines whether mutation is going to happen in each individual. In this research, we choose Roulette wheel selection and uniform distribution as the methods of selection and mutation, respectively, since these two methods are most commonly used. Figure 3.2 shows how Roulette wheel selection works while Figures 3.3 and 3.4 suggest how single-point crossover and mutation are implemented.



Individual	Fitness
1	1.0
2	2.0
3	3.0
4	4.0
5	5.0

Figure 3.2. Example of Roulette wheel selection. Source: [33] The idea of roulette wheel selection comes from a casino game, roulette, while the area of each portion is based on the score of the corresponding individual instead of being equally allocated. In the pie chart,the colored segments represent five individuals, where each occupies $\frac{\text{fitness score}}{\text{sum of all fitness scores}}\%$ of the chart.

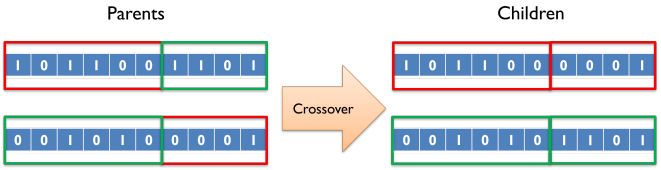


Figure 3.3. Crossover in GA. Adapted from: [32]. This figure presents an example of single-point crossover. On the left, the two bit strings are the original chromosomes. Once the point is randomly chosen, both chromosomes are divided into two parts, red and green, as shown. Next, the two offspring chromosomes are created as shown on the right by recombining genetic sections with those of the same color on the left. In this process, the position of each gene remains the same, meaning that the bit strings are exchanged section-by-section without moving the original sequence around.

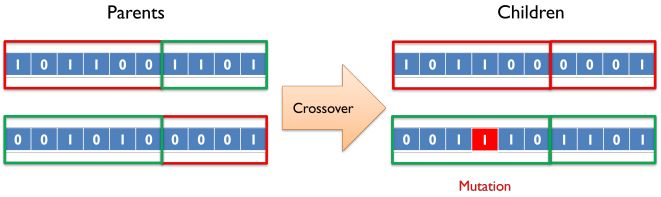


Figure 3.4. Mutation in GA. Adapted from: [32]. Under a certain stochastic distribution, mutation randomly happens in a population. This figure is only an example to show how an individual's chromosomes mutate based on Figure 2.3. The red block on the lower right bit string is the place where mutation occurs. In the red block, the original gene, 0, now is replaced by 1.

3.2.2 Fitness Function

In the cycle of GA, as shown in Figure 3.1, the other key component is evaluation. The core of evaluation is the fitness function, and it keeps the population evolving every cycle. In this research, we want to know which columns of W and corresponding rows of H in equation 3.9 contribute to the input matrix, X, since bias exists in the result of NNSVD-LRC NMF

between X and X_p. Thus, the fitness function is

$$\|X - W_g H_g\|_F \tag{3.13}$$

where W_g and H_g are structurally the same as W and H. However, values in the matrices are modified as mentioned in section 3.2.1. In this scenario, the entire process is likely to minimize the score evaluated from the fitness function evaluation.

3.2.3 An Example of GA

The following is an example of how GA works with NNSVD-LRC and NMF using the ALS update rule.

Let $X = \begin{bmatrix} 17 & 2 & 4 & 3 & 14 & 16 \\ 19 & 6 & 20 & 9 & 1 & 15 \\ 3 & 11 & 20 & 19 & 17 & 8 \\ 19 & 20 & 10 & 16 & 19 & 14 \\ 13 & 20 & 17 & 20 & 14 & 4 \end{bmatrix}$, whose rank is 5. By NNSVD-LRC NMF, we have

$W = \begin{bmatrix} 22.1010 & 12.1221 & 0 & 0 & 4.2339 \\ 28.5764 & 8.5667 & 9.7022 & 0 & 0 \\ 32.3477 & 0 & 1.9286 & 6.2586 & 0 \\ 39.8158 & 1.5369 & 0 & 0 & 5.6609 \\ 36.7520 & 0 & 0.1382 & 5.4692 & 0 \end{bmatrix}$ and

$H = \begin{bmatrix} 0.4255 & 0.3977 & 0.4406 & 0.4369 & 0.4083 & 0.3301 \\ 0.7454 & 0 & 0 & 0 & 0 & 0.6666 \\ 0 & 0 & 0.9952 & 0.0967 & 0 & 0.0170 \\ 0 & 0.5774 & 0.2948 & 0.7326 & 0.2072 & 0 \\ 0.0146 & 0.3411 & 0 & 0 & 0.9399 & 0 \end{bmatrix}$. However,

$WH = \begin{bmatrix} 18.5023 & 10.2340 & 9.7382 & 9.6564 & 13.0036 & 15.3765 \\ 18.5459 & 11.3649 & 22.2471 & 13.4241 & 11.6681 & 15.3087 \\ 13.7650 & 16.4788 & 18.0178 & 18.9052 & 14.5045 & 10.7112 \\ 18.1710 & 17.7661 & 17.5442 & 17.3963 & 21.5780 & 14.1684 \\ 15.6392 & 17.7746 & 17.9441 & 20.0780 & 16.1393 & 12.1348 \end{bmatrix}$, where the error between WH and X is 25.18714121.

Apply GA and create a set of population whose individuals are all bit strings. These bit strings are created to encode each column of W and corresponding row of H.

For instance, $\begin{bmatrix} 1 & 0 & 0 & 1 & 0 \end{bmatrix}$ stands for $W_g = \begin{bmatrix} 22.1010 & 0 & 0 & 0 & 0 \\ 28.5764 & 0 & 0 & 0 & 0 \\ 32.3477 & 0 & 0 & 6.2586 & 0 \\ 39.8158 & 0 & 0 & 0 & 0 \\ 36.7520 & 0 & 0 & 5.4692 & 0 \end{bmatrix}$ and $H_g =$

$$\begin{bmatrix} 0.4255 & 0.3977 & 0.4406 & 0.4369 & 0.4083 & 0.3301 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5774 & 0.2948 & 0.7326 & 0.2072 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

In addition, the fitness function is $\|X - W_g H_g\|_F$, where W_g and H_g are modified from W and H, accordingly.

During the GA process, every individual has its own rank. And an individual with a smaller fitness value gets a higher rank. In the meantime, each individual is scaled by $1/\sqrt{rank}$ as an input of the selection function. As we use the Roulette wheel selection function, an individual occupies an area according to its scale score. In this case, there are $\lfloor \frac{5}{2} \rfloor + 1 = 3$ individuals in a population. Hence, the scale scores of these three individual are $\frac{1}{\sqrt{1}} = 1$, $\frac{1}{\sqrt{2}} \approx 0.707$, and $\frac{1}{\sqrt{3}} \approx 0.577$. Then, the probability that the first one will be selected is $\frac{1}{1+0.707+0.577} \approx 43.8\%$, $\frac{0.707}{1+0.707+0.577} \approx 30.9\%$ for second one, and $\frac{0.577}{1+0.707+0.577} \approx 25.3\%$ for the remaining one. Indeed, every individual will be selected to avoid the duplicate chromosomes in this case.

Last, GA yields $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix}$. Meanwhile, $\|X - W_g H_g\|_F = 25.18714121$. We can notice that the results conserve all the columns and rows. The reason for this is that the size of this example is relatively small, and no column and row can be dropped.

■

After GA, $W_g H_g$ is now close to the input matrix, X.

As the frequencies of blue whale B calls are constant, we can apply the following algorithm

to analyze and detect the harmonic structure.

3.3 Harmonic Product Spectrum

The Harmonic Product Spectrum (HPS) is one of the algorithms that can find fundamental frequency with less computational cost because they do not take phase information into account. Indeed, such algorithms become unreliable if the fundamental frequency of signal is inconsistent. However, HPS is applicable in this research since blue whale calls are a repetition of the same frequency pattern.

HPS measures the fundamental frequency, $\tilde{Y}$, of the input spectrum, $X(\omega)$, through the equations:

$$Y(\omega) = \prod_{i=1}^{I} |X(i\omega)| \tag{3.14}$$

$$\tilde{Y} = \max_{\omega_i}(Y(\omega_i)) \tag{3.15}$$

where I is the maximum number of harmonics, ω is frequency range to be considered, and $Y(\omega)$ is the product of the down-sampled input spectrum according to De La Cuadra, Master, and Sapp's work [11]. Figure 3.5 shows how $Y(\omega)$ is executed. In addition, it should be noted that the octave error is common for HPS. To fix this problem, [11] suggests applying the following rule: "IF the second peak amplitude below initially chosen pitch is approximately 1/2 of the chosen pitch AND the ratio of amplitudes is above a threshold (e.g., 0.2 for 5 harmonics), THEN select the lower octave peak as the pitch for the current frame."

With HPS, we can detect blue whale B calls since the harmonic structure of the calls consist of $B \approx 15$ Hz, $B_2 \approx 30$ Hz, and $B_3 \approx 45$ Hz [3]. In other words, the fundamental frequency, 15 Hz, can be found when $I = 3$ in equation 3.14.

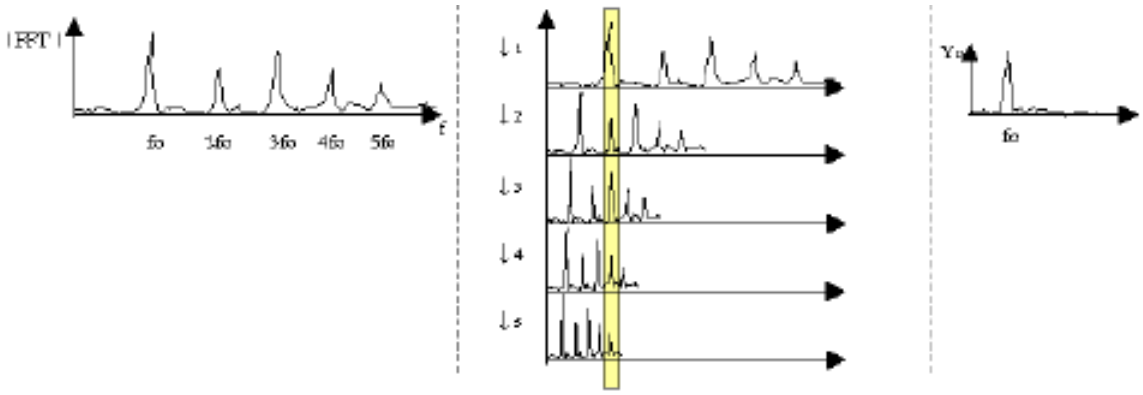


Figure 3.5. Implementation of HPS. Source: [11]. Assume that there is an input spectrum with five harmonics on the left of the figure. HPS down-samples the input spectrum with 5 different sampling rates from 1 to 5 and gets the five squeezed spectrums as the middle part of the figure. Therefore, any frequency, f, in original spectrum is now $\frac{f}{i}$, where i is the corresponding sampling rate. In order to process the multiplication of spectrums, HPS pads zero amplitudes behind those squeezed spectrums and makes them the same length since the spectrums are now in different sizes. Next, the algorithm multiplies the amplitudes of these five spectrums as the inner product of the multiplication of vectors and gets the product of these squeezed spectrums, which shown is on the right side of the figure. Last, HPS identifies the fundamental frequency, f_0, by looking for the maximum peak of the result spectrum.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Auto Detection of Blue Whale Calls

In this chapter, the performance of the algorithms is presented in the next four sections. The following tests are all run on a laptop (AMD Ryzen 7 5800H CPU @3.2GHz 16GB RAM) with MATLAB R2021b. The primary aim of this paper is to detect blue whale *B* calls since these calls hold a significant characteristic of harmonic structure and are easy to recognize in the spectrogram. To recognize them, first we factorize the input data into the W and H matrices, which respectively carry frequency and time information. Next, GA filters out those columns and corresponding vectors which are negligible. Finally, HPS concludes this auto detection process by recognizing whether there are blue whale *B* calls in the specified time interval.

4.1 Data Analysis

First, we transform the recording data by using Short-Time Fourier Transform (STFT). Different from FFT, STFT only takes a short period for every calculation. Therefore, the window function, length of window, FFT length, and overlapping rate are the parameters that STFT takes into account. The window length is less than or equal to the FFT length, and the remaining digits of the FFT length are padded with zeros (Zero Padding). According to [34], [35], the output of FFT achieves higher frequency resolution when zeros are added to the end of each segment. Notice that the having higher frequency resolution means having more points along the frequency axis. In other words, adding zeros does not improve the performance of the input signal, which is related to the limits of the equipment. The other element of STFT is the window function. The idea of the window function is to suppress the discontinuity on the starting and end points of each segment since FFT yields a broad frequency band to fit these discontinuous points. There are many types of window functions, and the one that works the best depends on the signal a researcher deals with. In general, Hanning (Hann) and Hamming windows are the most popular functions. They have a similar structure, while their coefficients are slightly different. A generalized Hamming window

function is written as

$$w(n) = a + (1 - a)cos\left[\frac{2\pi(n - \frac{N}{2})}{N}\right] \tag{4.1}$$

where $a < 1$, N is the length of the window, and $n = 1, 2, ..., N$. And $w(n)$ is considered the Hamming window function as $a = 0.54$, while it is regarded as the Hanning (Hann) window function when $a = 0.5$. In this paper, we have the Hamming window function with a sampling rate as the window length since it has lower sidelobes than the Hanning window according to [34]. For FFT length, as the computers process in binary, length in $2^n \quad (n \in \mathbb{N})$ improves the efficiency of computation according to [36]. Here, we have FFT length as 2^{R+1}, where R is the number that makes 2^R close to F_s, which is the data sampling rate. In other words, FFT length is $2^{12+1} = 16384$ as the sampling rate is 8,000 per second for the vector sensors and $2^{12} = 8192$, which is the number closest to 8,000. The reason is that this number helps us get the frequency resolution approximately of 0.5 Hz. Last, it is important to consider the overlapping rate. In Figure 4.1, each window overlaps the previous one. This process helps researchers see more closely how frequencies change with time. In this thesis, it is unnecessary to specifically set the overlapping rate since the pattern of blue whale B calls are repetitive and consistent with the frequency. Therefore, to reduce the computational cost and make the time measuring easier for the following experiments, each 1-second time frame is calculated by applying FFT with window length of 16,384, and the remaining length of 8,384 is padded with zeros.

In order to obtain the input matrix as a non-negative matrix, we make an adjustment which takes the minimum intensity, denoted as I'_0, as reference to equation 2.4. This guarantees $\mathrm{dB}' = 10\log_{10}\frac{I}{I'_0} \geq 0$ since $\frac{I}{I'_0} \geq 1$. With all the parameters just identified, we can obtain the spectrogram. In Figure 4.2, we only show the frequency range from 0 to 100 Hz to highlight the blue whale B calls. It is a 30-minute recording collected on August 3, 2021. The calls are those consisting of frequencies at $B \approx 15$ Hz, $B_2 \approx 30$ Hz, and $B_3 \approx 45$ Hz [3].

In next two sections, we focus on the spectrogram shown in Figure 4.2.

4.2 NMF Results

With the non-negative spectrogram shown in Figure 4.2, we can apply NNSVD-LRC NMF. Considering that GA will take most of the computational resources, the input spectrogram

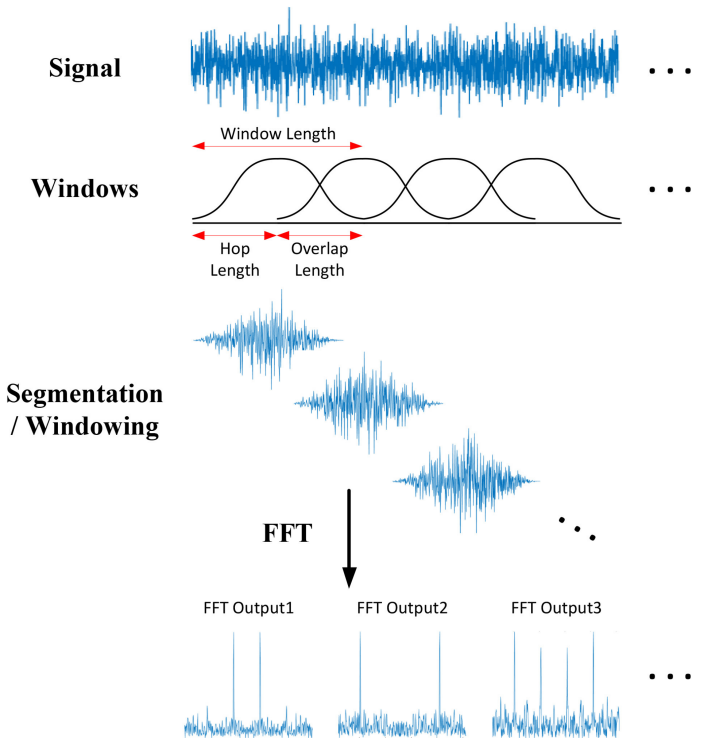


Figure 4.1. Short-Time Fourier Transform. Source: [37]. STFT first divides an input signal into segments, according to the window size. Overlapping is recommended in order to see how signal change with time. Next, each segment is convoluted with certain type of window function, depending on the signal. Last, we apply FFT to each segment and obtain the frequency information from each time segment.

is constrained to the frequency range from 0 to 100 Hz. Figure 4.3 is an example that shows the W and H matrices of the time interval between 960 and 1020 seconds by applying NNSVD-LRC NMF with the ALS update rule while the number of the column vector of W and the row vector of H is 61, since the input spectrogram is a matrix with a rank of 61. In addition, Figure 4.4 illustrates both the original spectrogram and the one obtained by multiplying the W and H matrices. By looking at the colorbars, we can see the magnitudes are different since NNSVD-LRC NMF with ALS is a low-rank approximation. Specifically, the difference between the two spectrograms is 1.1093×10^3. Nevertheless, the features remain approximate results, such as the blue whale B call between 1010 and 1020 seconds.

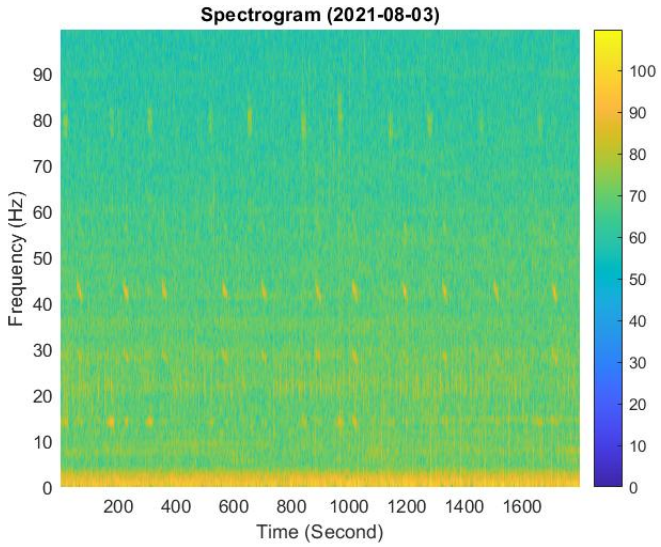


Figure 4.2. Recording data on August 3, 2021. This is a 30-minute recording. To highlight the blue whale calls, this spectrogram only shows the frequency range up to 100 Hz.

Moreover, Figure 4.5 highlights the blue whale call's features in the W and H matrices by only showing the columns and rows numbered 1 to 10, and the second and fourth columns of the W matrix, while corresponding rows of the H matrix are most likely where the blue whale B call is.

Next, the accuracy, duration, and number of iterations of NMF are examined. Particularly, we notice how MU and ALS with NNSVD-LRC perform in the first block of time with different time intervals in the input spectrogram. Here, the time interval is not the same as the window length that we apply to FFT. The input spectrogram is shown in Figure 4.2. The number of columns of W and corresponding rows of H is the rank of input spectrogram, and we find out that applying ranks in even numbers will cause singularity for ALS with NNSVD-LRC since a linear equation will be solved according to the ALS update rule. We, therefore, only apply the time interval with an odd number to prevent the singular problem. The stop criterion is either when the difference of errors between the current iteration and the previous one is less than 10^{-4} or the number of iterations meets 10,000. In Table 4.1

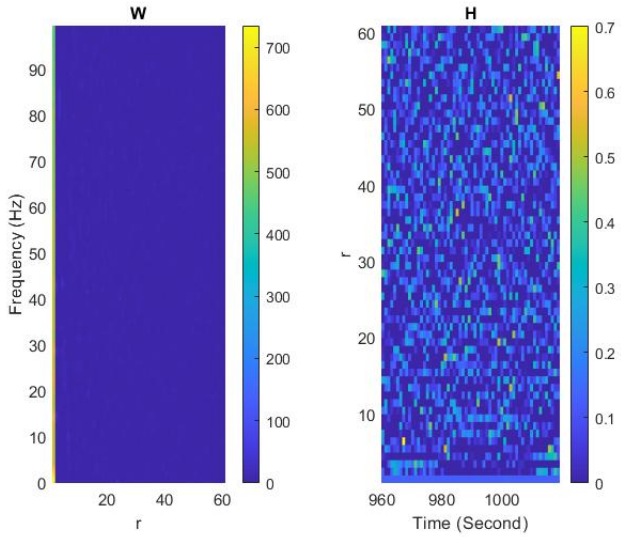


Figure 4.3. W and H matrices produced by NNSVD-LRC NMF. The W matrix on the left contains frequency information of input data in each column while the H matrix shows the time information in the corresponding rows.

(shown later in this chapter), the values of the Frobenius norm of both update rules increase as the time interval gets larger. However, the duration and number of iterations between ALS and MU have a significant difference. First, the number of iterations of ALS remain two regardless of the time interval, while there is no pattern for MU that suggests a relation between the time interval and the number of iterations. Second, the values of the Frobenius norm for MU are smaller than those for ALS. Last, the duration of MU changes with the size of the time interval but not for ALS. Thus, we can see that ALS using the initialization meets the local minimum faster than MU, even though MU is closer to the global minima. This result validates what we discussed in section 3.1.2, which was that MU takes more iterations to converge [28], [29].

With these results, the object of GA is to filter out the vector spaces that cause the error between the NNSVD-LRC NMF result and the original spectrogram in the W and H matrices.

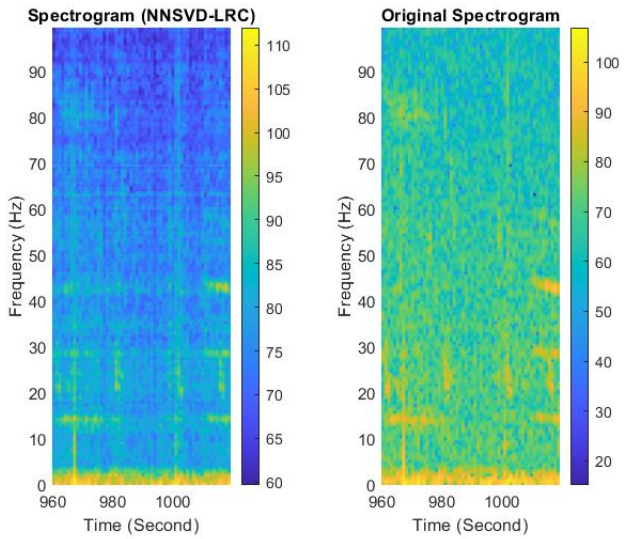


Figure 4.4. Comparison of original and NNSVD-LRC NMF spectrograms. The features in the original spectrogram on the right remain in the spectrogram on the left even though the magnitudes are different.

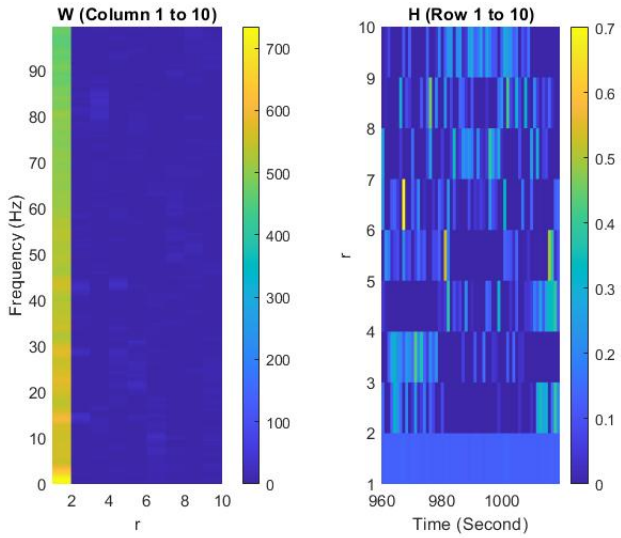


Figure 4.5. Magnified W and H matrices. The second column of the W matrix on the left contains the frequencies B, B_2, and B_3 of blue whale calls while the fourth column holds B_2, and B_3. Compared to the original spectrogram in Figure 4.4, the second and fourth rows of the H matrix on the right have a similar period of blue whale calls.

Table 4.1. NMF Results by Using MU and ALS with NNSVD-LRC.

		MU			ALS		
Time Interval (s)	Rank	$\|X - WH\|_F$	# of Iterations	Duration (s)	$\|X - WH\|_F$	# of Iterations	Duration (s)
11	11	163.2149	734	0.074655	262.9029	2	0.025782
21	21	213.8729	688	0.101032	437.0081	2	0.025620
31	31	243.6288	885	0.128251	602.4804	2	0.017756
41	41	267.1358	920	0.174437	773.5125	2	0.013496
51	51	284.5322	1061	0.230870	941.0571	2	0.012907
61	61	295.7271	1146	0.79175	1.1191×10^3	2	0.030516
71	71	303.6512	940	0.301869	1.2873×10^3	2	0.014283
81	81	305.1922	1020	0.416658	1.4529×10^3	2	0.012245
91	91	306.6802	1246	0.537783	1.6254×10^3	2	0.014735
101	101	308.7541	851	0.440325	1.7949×10^3	2	0.013365

4.3 GA Result

In order to filter out the insignificant vectors, GA holds a set of bit strings, an initial population, and determines which string represents output, W_g and H_g, the best by evolving the population according to the GA cycle. Figure 4.6 presents an example that is a continuation of the one in Figure 4.3. The GA spectrogram keeps most features of the original spectrogram even though the whale call between 962 and 968 seconds does not exist in the original one. Furthermore, Figure 4.7 shows that only six vectors remain in W_g and H_g using GA and the Frobenius norm between the GA and the original spectrograms is now 640.6305, which is almost half of the original error.

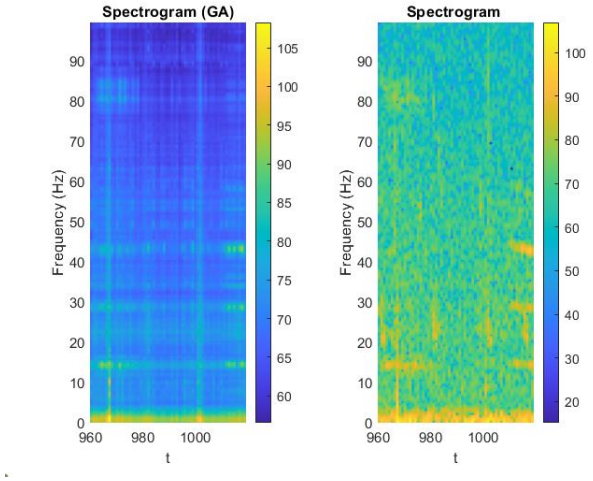


Figure 4.6. Comparison of GA and original spectrograms. After filtering out less significant vectors, GA yields the spectrogram on the left. Most features in the original spectrogram are kept in the GA one, while some distortion remains. The restoration has the blue whale B calls not only between 1010 and 1020 seconds but also between 962 and 968 seconds. This error is made by the NNSVD-LRC NMF with the ALS update rule as shown in Figure 4.4, where we can see the fake B call between 962 and 968 seconds.

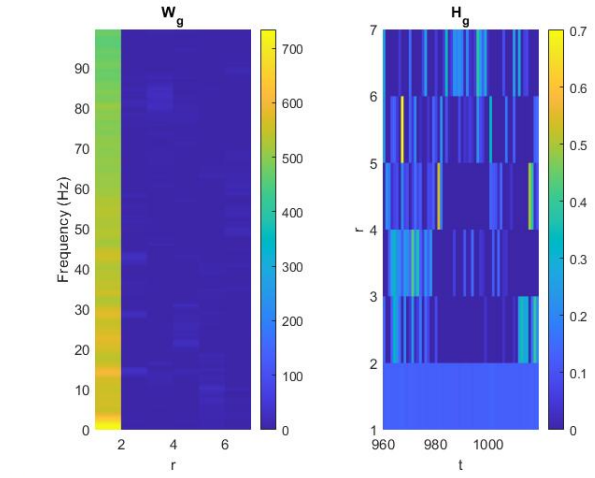


Figure 4.7. W_g and H_g matrices. After the GA process, only six vectors are left in W_g and H_g. And the error between the GA and the original spectrograms is nearly half of NNSVD-LRC NMF result.

We see how GA reduced the number of vectors and the error. Next, we need to check how GA works with different update rules of NMF. In the following experiment, we take the result from Table 4.1, apply GA, and see how it goes. The duration of GA process, number of remaining vectors, and error will be considered. The size of population is set to be two times of the number of vectors.

In Table 4.2, MU does not show any change; the errors and number of remaining vectors are the same as before the GA process. Nonetheless, the errors are notably decreased due to the number of remaining vectors. In other words, GA eliminates the most vectors in the ALS result to make the result close to the original spectrogram, and we discover that the remaining vectors are almost first few vectors. The reason for that is the way we initialize W and H, which uses truncated SVD and sequentially assign the values. This makes the vectors in first few columns of W or corresponding rows of H more significant. Overall, GA does not affect the result of MU. Therefore, we only apply GA to the ALS in the following section.

Table 4.2. GA Results with MU and ALS.

	MU			ALS		
Time Interval (s)	$\|X - W_g H_g\|_F$	# of Remaining Vectors	Duration (s)	$\|X - W_g H_g\|_F$	# of Remaining Vectors	Duration (s)
11	163.2149	11	0.031855	249.4287	6	0.036562
21	213.8729	21	0.064781	363.1719	6	0.127663
31	243.6288	31	0.165211	447.7495	6	0.198504
41	267.1358	41	0.268938	518.517	6	0.68683
51	284.5322	51	0.401972	579.7843	6	0.967323
61	295.7271	61	0.515081	636.4363	7	1.467205
71	303.6512	71	0.793914	688.6056	7	1.998085
81	305.1922	81	1.181693	735.9268	6	2.421773
91	306.6802	91	1.291844	780.1870	7	3.383358
101	308.7541	101	1.730327	819.6323	7	5.135152

4.4 Performance and Analysis

In this section, we review all the algorithms we mentioned in Chapter 3 to see how accurately this system can recognize the blue whale B calls. We include two 30-minute recording data sets, which are shown in Figures 4.8 and 4.9. In Figure 4.8, blue whale B calls are relatively easier to visualize than those in Figure 4.9 since the frequency of noise overlaps the B1 and B2 frequencies of blue whale calls in Figure 4.9. Therefore, in this test, we can see how noise can affect the accuracy of detection of calls.

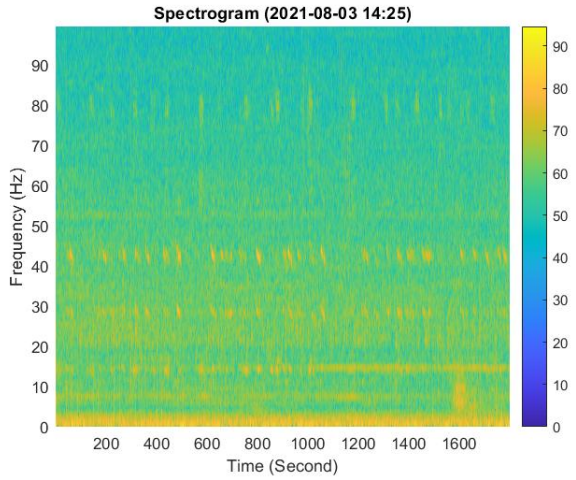


Figure 4.8. Test Spectrogram-1. Blue whale B calls in this spectrogram are relatively easy to visualize despite the consistent noise around 15 Hz between 1,000 and 1,800 seconds, which is unknown.

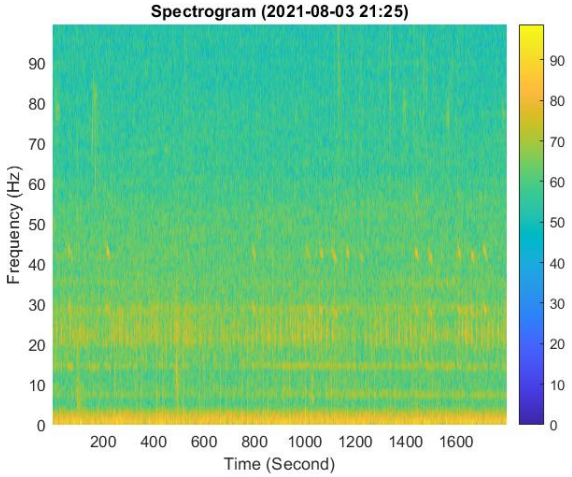


Figure 4.9. Test Spectrogram-2. Fewer blue whale B calls occur in this spectrogram, compared to Figure 4.8. In addition, noise around 15 Hz and between 20 and 30 Hz makes the harmonic structure of the blue whale B calls ambiguous.

First, we review the different time intervals and different update rules of NMF against the spectrogram in Figure 4.8 to see which interval and which update rule yield the best result. Then, we can apply this outcome to the spectrogram in Figure 4.9 and analyze the performance of the system. The system's recognition result is as shown in Figure 4.10, where "Y" stands for the existence of a B call, and "N" stands for no detection, and verified by visual analysis.

In statistic, we can evaluate the performance of a system by confusion matrix. In a confusion matrix, it indicates the sensitivity, specificity, false positive and false negative rates of the system [38]. In this thesis, the goal is to find the algorithms that yield the lowest percentage of type I error, which is least likely to trigger an alert when a whale call is non-existent. A type I error is when an investigation incorrectly rejects null hypothesis [38]. It is also called a false positive. Indeed, type II error is also crucial for an automatic detection system because it suggests the rate that a presence is missed by the detector. However, the reason for considering type I error is that we want to reduce the human cost for physically checking. So, we obtain confusion matrices to help us find out which intervals in each update rule meet

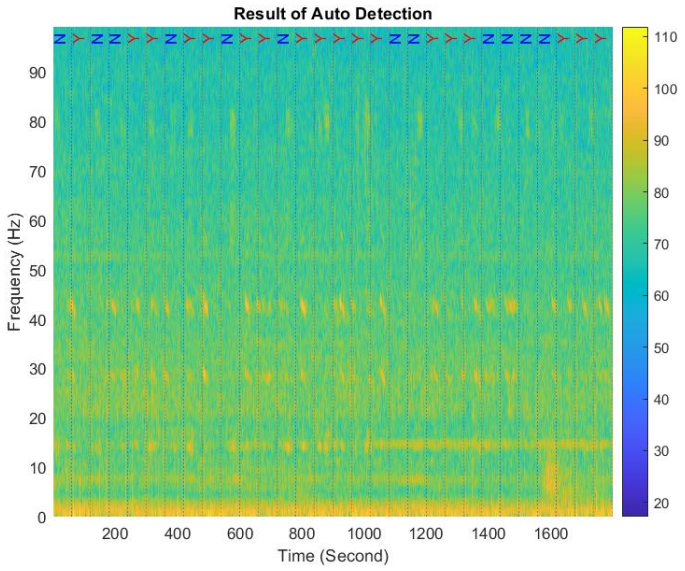


Figure 4.10. An example of the result of auto detection. "Y"s in red stand for detection of blue whale B calls, while "N"s stand for no detection. In this example, the time interval is 61 seconds, and the update rule is ALS with GA.

the requirement. As shown in Table 4.3, an 81-second interval has the lowest type I error rate for MU despite the specificity being 50%, which is caused by the insufficient sample size. In Table 4.4, the 71-second interval with the ALS and the GA has the lowest type I error rate and second highest level of sensitivity. Thus, we apply MU with an 81-second interval and ALS and GA with a 71-second interval to the spectrogram in Figure 4.9.

From Table 4.5, we can see the sensitivity of MU is higher than that of the ALS with GA, while the percentage of type I error is also much higher with MU. Moreover, MU takes less time to finish the process. Thus, we can conclude that ALS with GA can alert us to the detection of blue whale calls more accurately than MU despite it taking more time to finish the calculation. Furthermore, the noise in the second test affects the result of both update rules since it interferes with the harmonic structure by overlapping with a broadband frequency. This will need further research to overcome noise interference.

Table 4.3. Performance of Auto Detection with MU.

Time Interval = 11 Seconds			
		Auto Detection	
		Yes	No
Actual	Yes	17	27
	No	15	121
Duration = 16.764 Seconds			

Time Interval = 21 Seconds			
		Auto Detection	
		Yes	No
Actual	Yes	28	10
	No	9	43
Duration = 15.541 Seconds			

Time Interval = 31 Seconds			
		Auto Detection	
		Yes	No
Actual	Yes	25	7
	No	9	19
Duration = 17.033 Seconds			

Time Interval = 41 Seconds			
		Auto Detection	
		Yes	No
Actual	Yes	31	1
	No	7	6
Duration = 15.631 Seconds			

Time Interval = 51 Seconds			
		Auto Detection	
		Yes	No
Actual	Yes	23	2
	No	4	7
Duration = 18.119 Seconds			

Time Interval = 61 Seconds			
		Auto Detection	
		Yes	No
Actual	Yes	23	1
	No	3	3
Duration = 17.757 Seconds			

Time Interval = 71 Seconds			
		Auto Detection	
		Yes	No
Actual	Yes	21	1
	No	2	2
Duration = 21.029 Seconds			

Time Interval = 81 Seconds			
		Auto Detection	
		Yes	No
Actual	Yes	19	1
	No	1	1
Duration = 22.411 Seconds			

Time Interval = 91 Seconds			
		Auto Detection	
		Yes	No
Actual	Yes	16	1
	No	2	1
Duration = 23.419 Seconds			

Time Interval = 101 Seconds			
		Auto Detection	
		Yes	No
Actual	Yes	15	0
	No	1	2
Duration = 26.649 Seconds			

Table 4.4. Performance of Auto Detection with ALS.

Time Interval = 11 Seconds			
		Auto Detection	
		Yes	No
Actual	Yes	13	31
	No	7	129
Duration = 14.917 Seconds			

Time Interval = 21 Seconds			
		Auto Detection	
		Yes	No
Actual	Yes	21	17
	No	3	49
Duration = 24.478 Seconds			

Time Interval = 31 Seconds			
		Auto Detection	
		Yes	No
Actual	Yes	17	15
	No	3	25
Duration = 35.641 Seconds			

Time Interval = 41 Seconds			
		Auto Detection	
		Yes	No
Actual	Yes	15	17
	No	0	13
Duration = 50.187 Seconds			

Time Interval = 51 Seconds			
		Auto Detection	
		Yes	No
Actual	Yes	13	8
	No	2	9
Duration = 78.997 Seconds			

Time Interval = 61 Seconds			
		Auto Detection	
		Yes	No
Actual	Yes	17	7
	No	1	5
Duration = 87.532 Seconds			

Time Interval = 71 Seconds			
		Auto Detection	
		Yes	No
Actual	Yes	16	6
	No	0	4
Duration = 106.436 Seconds			

Time Interval = 81 Seconds			
		Auto Detection	
		Yes	No
Actual	Yes	10	10
	No	0	2
Duration = 122.229 Seconds			

Time Interval = 91 Seconds			
		Auto Detection	
		Yes	No
Actual	Yes	12	5
	No	0	3
Duration = 189.393 Seconds			

Time Interval = 101 Seconds			
		Auto Detection	
		Yes	No
Actual	Yes	11	4
	No	1	2
Duration = 227.96 Seconds			

Table 4.5. Comparison of the Performance of MU and ALS with GA.

MU			
Time Interval = 81 Seconds			
		Auto Detection	
		Yes	No
Actual	Yes	9	1
	No	6	7
Duration = 20.877 Seconds			

ALS with GA			
Time Interval = 71 Seconds			
		Auto Detection	
		Yes	No
Actual	Yes	6	7
	No	0	13
Duration = 106.877 Seconds			

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Conclusion and Future Work

In this thesis, the goal has been to create a system that could help people track the activities of blue whales by detecting and analyzing their calls. We describe the regulations and importance of protecting marine mammals in Chapter 1 while we also introduced basic knowledge about blue whale calls and several related research works. In Chapter 2, we discussed the data collection in Monterey Bay area and underwater acoustics. To set up our proposed system, we applied NMF, GA, and HPS, which were discussed in Chapter 3. Finally, the system's performance was examined in Chapter 4.

Specifically, we applied NMF with specific initialization, NNSVD-LRC, and considered the behavior of two different update rules, ALS and MU. ALS converged to the local minimum relatively quickly but yielded a higher error than MU. Thus, GA was adopted for our system to reduce the error of ALS by reducing the column and corresponding row spaces of the W and H matrices, respectively. Last, the HPS examined every columns of the W matrix to see whether the existence of blue whale B call is true. The structure of this system is shown in Figure 5.1. By analyzing the system's performance on same data using a different time frame size, we drew a preliminary conclusion that ALS with GA has the lowest type I error rate when the time interval is 71 seconds time interval, while MU's lowest type I error rate it is achieved with a 81-second interval. Next, we wanted to see how these two algorithms act on the data that was recorded in a noisy environment, and we found ALS with GA has the lowest percentage of type I error. Indeed, MU has more sensitivity than ALS with GA, but that sensitivity leads to more false alarms that then need to be checked by humans. Consequently, the system with fewer false alarms can effectively reduce the strain on human resources.

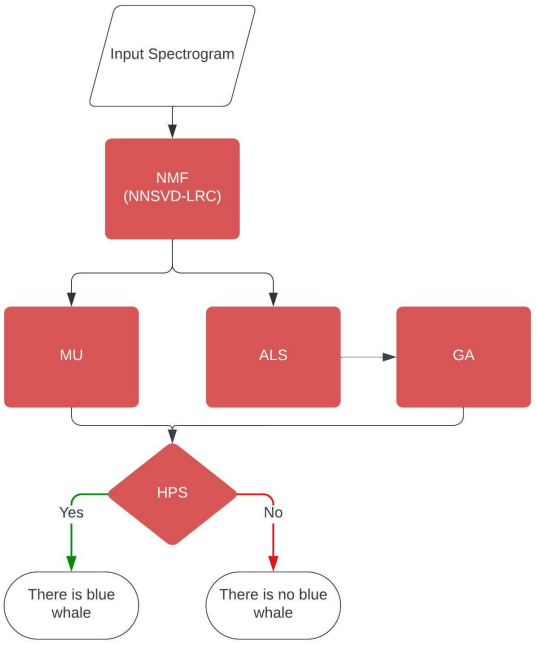


Figure 5.1. Auto detection algorithm. This algorithm begins with an input spectrogram. Next, the NMF factorizes this spectrogram into W and H matrices with NNSVD-LRC initialization. In the meantime, the results vary because of the update rules, which are ALS and MU. Therefore, the algorithm includes GA to reduce the error, which is created by the approximation. Nevertheless, the result of MU with GA does not change. So, the algorithm only obtains ALS with GA. Finally, HPS examines the column space of the W matrix, which is produced by either ALS with GA or MU, and determines whether the blue whale call occurs.

In conclusion, NNSVD-LRC NMF with ALS and GA can accurately alert researchers to the existence of blue whale B calls even though the system's sensitivity for detection is only average. By comparison, MU has greater sensitivity for detection, but this system may not be the preferred choice for detecting marine mammals when the environment is noisy as the probability of false alarm will also be high.

Future work extending this research could include the parallel computation and detection for

other types of marine mammals. As mentioned in Chapter 3, GA may not give the optimal solution since it is founded on a probability model. With parallel computation, GA can be executed on different nodes of a cluster, and the one which yields the best solution can then be used as input for HPS to process the detection of harmonic structures. Moreover, there are various marine mammals that we have not discussed in this thesis. By understanding the harmonic structure of those creatures' calls, we can apply this system to track their activities and recognize their behavior. In addition, we will use the auto detection algorithm developed in this thesis to build an application for the Taiwan Navy and utilize in different data set. Another direction of future research is to use vector sensor data to identify the blue whale locations.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX: MATLAB CODE

```
% main.m
clear
clf

% Input filename
filename = '';
[data, rate] = audioread(filename);
time_in_second = floor(length(data)/rate);
R_C = data(1:time_in_second*rate,1);
% Maximum frequency to compute
con_f = 500;
% Maximum frequency showing on spectrogram
show_f = 100;
% Update rule for NMF ('mu' or 'als')
method = 'als';
% Size of time frame
interval = 70;
%%%%FFT parameter%%%%%%
% Window Function
WS = hamming(rate,'periodic');
%FFT length
FL = 16384;
con_f = con_f*FL/rate;
show_f = floor(show_f*FL/rate);
f_res = rate/FL;
%%%%%%STFT%%%%%%%%
[s,f,t] = stft(R_C,rate,'Window',WS,'OverlapLength',...
0,'FFTLength',FL,'FrequencyRange','onesided');
mag = 2 * abs(s)/FL;
ref = min(min(mag(1:con_f,:)));
db = 20*log10(mag(1:con_f,:)/ref);
n_f = f(1:show_f);

pcolor(t,n_f,db(1:show_f,:));
shading flat
```

```
colorbar
title('Result of Auto Detection')
xlabel('Time (Second)')
ylabel('Frequency (Hz)')
hold on

% Maximum number of harmonic structure. Here, 3rd harmonic frequency has the
% strongest intensity
max_har = 3;
interval_count = zeros(ceil(time_in_second/interval),1);
tic
for i = 1:interval:time_in_second
    check_flag = 0;
    if i+interval < size(db,2)
        % NNSVD-LRC
        [Wnnd, Hnnd, p] = NNSVD_LRC(db(:,i:i+interval),method);
        % GA
        [W, H, x] = GA(Wnnd, Hnnd, db(:,i:i+interval));
    % Last time frame may not be integer
    else
        [Wnnd, Hnnd, p] = NNSVD_LRC(db(:,i-1:end),method);
        [W, H, x] = GA(Wnnd, Hnnd, db(:,i-1:end));
    end
    % Delete the columns and corresponding rows that are seemed to be less
    % significant
    del = 0;
    for l = 1:size(x,2)
        if x(l) == 0
            W(:,l+1-del) = [];
            H(l-del+1,:) = [];
            del = del + 1;
        end
    end

    interval_num = ((i-1)/interval) + 1;
    disp(interval_num);
    % Using HPS to check the harmonic structure
    for r = 1:size(W,2)
        if check_flag == 0
            hps = W(:,r);
```

```
            Yi = hps;
            for j = 2:max_har
                ds = downsample(Yi,j);
                hps = [ds; zeros(size(W(:,r),1)-size(ds,1),1)].*hps;
            end
            avg = mean(hps,"all");
            [Value,PEAK] = findpeaks(hps,"SortStr","descend","NPeaks",2);
            PEAK = PEAK.*f_res;
            if size(PEAK,1) > 1
                if (Value(2,1) >= Value(1,1)*0.45) && (Value(2,1) <= Value(1,1)*0.55)..
                 && (PEAK(2,1) < PEAK(1,1))
                    k = 2;
                else
                    k = 1;
                end
                if PEAK(k,1) > 14 && PEAK(k,1) < 16 && r ~= 1
                        interval_count(interval_num) = 1;
                        check_flag = 1;
                end
            end
        end
    end
end
toc
% Show "Y" or "N" on the spectrogram in each time frame
for i = 1:size(interval_count,1)
    if interval_count(i) == 1
        xl = xline((i*interval)-1,'r','Y');
        xl.LabelHorizontalAlignment = 'left';
    else
        xl = xline((i*interval)-1,':b',' ');
        xl.LabelHorizontalAlignment = 'left';
    end
end
hold off

% NNSVD-LRC.m

function [W,H,r] = NNSVD_LRC(input, method)
    input_size = size(input);
```

```
116     r = rank(input);
117     p = floor(r/2+1);
118     [U,S,V] = svds(input,p);
119     S = S.^0.5;
120     Y = U*S;
121     Z = S*V';
122     Xp = Y*Z;
123     Wi = zeros(input_size(1),r);
124     Hi = zeros(r,input_size(2));
125     % Assign data
126     Wi(:,1) = abs(Y(:,1));
127     Hi(1,:) = abs(Z(1,:));
128     j = 2;
129     for i = 2:r
130         if mod(i,2) == 0
131             Wi(:,i) = max(Y(:,j),0);
132             Hi(i,:) = max(Z(j,:),0);
133         else
134            Wi(:,i) = max(-Y(:,j),0);
135            Hi(i,:) = max(-Z(j,:),0);
136            j = j+1;
137         end
138     end
139     % NMF process
140     [W,H] = nnmf(Xp, r,'w0',Wi,'h0',Hi,'algorithm',method,'Options',statset(...
141     'Display', 'final','MaxIter',10000));
142 end
143
144
145 % GA.m
146
147 function [W_GA, H_GA,x] = GA_V3(W_origin, H_origin, origin)
148     W = W_origin;
149     H = H_origin;
150
151     populationsize = size(W,2)*2;
152     nvar = size(W,2)-1;
153     opts = optimoptions('ga', ...
154         'PopulationType', 'bitstring',...
155         'PopulationSize', populationsize,...
```

```
        'CreationFcn', @gacreationuniform,...
        'SelectionFcn', @selectionroulette,...
        'FitnessScalingFcn',@fitscalingrank,...
        'MutationFcn', @mutationuniform,...
        'CrossoverFcn', @crossoversinglepoint);
    func = @(chrom)GA_obj_v3(chrom, W, H, origin);
    x = ga(func, nvar, opts);
    for j = 1:length(x)
        if x(j) == 0
            W(:,j+1) = 0;
            H(j+1,:) = 0;
        end
    end
    W_GA = W;
    H_GA = H;
end

% GA_obj.m

function fitn = GA_obj(chrom, W, H, X)
    fitn = zeros(1,size(chrom,1));
    for i = 1:size(chrom,1)
        phen_W = W;
        phen_H = H;

        %%% Remove used vector spaces from phenotype
        for j = 1:size(chrom,2)
            if chrom(i,j) == 0
                phen_W(:,j+1) = 0;
                phen_H(j+1,:) = 0;
            end
        end
        %%% Use Fro norm to estimate the distance of orginal matrix and
        %%% estrimated matrix. The goal is to get the minimum distance
        %%% after combination.
        x_bar = (phen_W*phen_H);
        x_bar = X - x_bar;
        fitn(1,i) = norm(x_bar,"fro");
    end
end
```

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

[1] National Oceanic and Atmospheric Administration Fisheries.(1972, Oct. 21), "Marine mammal protection act policies, guidance, and regulations." Accessed: Aug. 14, 2023 [Online]. Available: https://www.fisheries.noaa.gov/national/marine-mammal-protection/marine-mammal-protection-act-policies-guidance-and-regulations

[2] W. J. Richardson, C. R. Greene Jr, C. I. Malme, and D. H. Thomson, *Marine mammals and noise*. Cambridge, MA, USA: Academic Press, 2013, pp. 159–204.

[3] W. K. Oestreich *et al.*, "Animal-borne metrics enable acoustic detection of blue whale migration," *Current Biology*, vol. 30, no. 23, pp. 4773–4779.e3, Dec. 2020 [Online]. Available: https://doi.org/https://doi.org/10.1016/j.cub.2020.08.105

[4] K. Chung, A. Sutin, A. Sedunov, and M. Bruno, "Demon acoustic ship signature measurements in an urban harbor," *Advances in Acoustics and Vibration*, vol. 2011, May 2011 [Online]. Available: https://doi.org/10.1155/2011/952798

[5] J. P. Ryan *et al.*, "Oceanic giants dance to atmospheric rhythms: Ephemeral wind-driven resource tracking by blue whales," *Ecology Letters*, vol. 25, no. 11, pp. 2435–2447, 2022.

[6] W. K. Oestreich, B. Abrahms, M. F. McKenna, J. A. Goldbogen, L. B. Crowder, and J. P. Ryan, "Acoustic signature reveals blue whales tune life-history transitions to oceanographic conditions," *Functional Ecology*, vol. 36, no. 4, pp. 882–895, 2022.

[7] F. Briggs, R. Raich, and X. Z. Fern, "Audio classification of bird species: A statistical manifold approach," in *2009 Ninth IEEE International Conference on Data Mining*. IEEE, 2009, pp. 51–60.

[8] W. Mu, B. Yin, X. Huang, J. Xu, and Z. Du, "Environmental sound classification using temporal-frequency attention based convolutional neural network," *Scientific Reports*, vol. 11, no. 1, p. 21552, 2021.

[9] T.-H. Lin, S.-H. Fang, and Y. Tsao, "Improving biodiversity assessment via unsupervised separation of biological sounds from long-duration recordings," *Scientific Reports*, vol. 7, no. 1, p. 4547, 2017.

[10] A. Kroon, "Comparing conventional pitch detection algorithms with a neural network approach," *arXiv preprint arXiv:2206.14357*, 2022.

[11] P. de la Cuadra, A. S. Master, and C. S. Sapp, "Efficient pitch detection techniques for interactive music," in *International Conference on Mathematics and Computing*, 2001 [Online]. Available: https://api.semanticscholar.org/CorpusID:18771948

[12] J. W. Foster, "Surface vessel acoustic signal direction of arrival estimation by vector sensor processing with the maximum eigengap estimator," M.S. thesis, Dept. of Operations Research, NPS, Monterey, CA, USA, 2021 [Online]. Available: https://hdl.handle.net/10945/67711

[13] J. Barry, L. Kuhnz, K. Buck, C. Lovera, and P. Whaling, "Potential impacts of the mars cable on the seabed and benthic faunal assemblages," *Monterey Bay Aquarium Research Institute*, 2008.

[14] L. D. Talley, G. L. Pickard, W. J. Emery, and J. H. Swift, "Chapter 7 - dynamical processes for descriptive ocean circulation," in *Descriptive Physical Oceanography (Sixth Edition)*, L. D. Talley, G. L. Pickard, W. J. Emery, and J. H. Swift, Eds. Cambridge, MA, USA: Academic Press, 2011, pp. 187–221 [Online]. Available: https://doi.org/https://doi.org/10.1016/B978-0-7506-4552-2.10007-1

[15] University of Rhode Island and Inner Space Center, "Hydrophone/ receiver dosits." Accessed: Jul. 19, 2023 [Online]. Available: https://dosits.org/galleries/technology-gallery/basic-technology/hydrophonereceiver/

[16] V. P. Ramachandran, D. Thomas, and T. Vinod, "Design and development of a broadband spherical hydrophone using pzt-4," *ISSS Journal of Micro and Smart Systems*, vol. 9, pp. 163–171, 2020.

[17] W. Kuperman and P. Roux, *Underwater Acoustics*. in Springer Handbook of Acoustics, T. Rossing, Ed., New York, NY, USA: Springer, 2007 [Online]. Available: https://doi.org/10.1007/978-0-387-30425-0_5

[18] GeoSpectrum Technologies Inc., *M20-105 Specification Sheet*, 2022-03. Accessed: Jul. 3, 2023 [Online]. Available: https://geospectrum.ca/wp-content/uploads/2022/03/M20-105-v8.pdf

[19] L. Brekhovskikh and Y. P. Lysanov, *Fundamentals of ocean acoustics*. Acoustical Society of America, 2004, pp. 1–34.

[20] F. B. Jensen, W. A. Kuperman, M. B. Porter, H. Schmidt, and A. Tolstoy, *Computational ocean acoustics*. New York, NY, USA: Springer, 2011.

[21] R. J. Urick, *Principles of underwater sound*. Peninsula publishing, 1983, pp. 19–30 [Online]. Available: https://cir.nii.ac.jp/crid/1130000795368781056

[22] D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," *Advances in neural information processing systems*, vol. 13, 2000.

[23] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Journal of Computational Statistics and Data Analysis*, vol. 52, no. 1, pp. 155–173, 2007.

[24] S. Fathi Hafshejani and Z. Moaberfard, "Initialization for non-negative matrix factorization: A comprehensive review," *International Journal of Data Science and Analytics*, vol. 16, no. 1, pp. 119–134, 2023.

[25] S. Theodoridis and K. Koutroumbas, *Pattern Recognition, Fourth Edition*. New York, NY, USA: Academic Press, 2009.

[26] S. M. Atif, S. Qazi, and N. Gillis, "Improved SVD-based initialization for nonnegative matrix factorization using low-rank correction," *Pattern Recognition Letters*, vol. 122, pp. 53–59, 2019.

[27] M. Kilmer, L. Horesh, H. Avron, and E. Newman, "Tensor-tensor products for optimal representation and compression," *arXiv preprint arXiv:2001.00046*, 2019.

[28] N. Gillis and F. Glineur, "Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization," *Neural Computation*, vol. 24, no. 4, pp. 1085–1105, 2012.

[29] N. Gillis and F. Glineur, "Nonnegative factorization and the maximum edge biclique problem," *arXiv preprint arXiv:0810.4225*, 2008.

[30] H. Liu, X. Li, and X. Zheng, "Solving non-negative matrix factorization by alternating least squares with a modified strategy," *Data Mining and Knowledge Discovery*, vol. 26, pp. 435–451, 2013.

[31] A. A. Gozali and S. Fujimura, "Localized island model genetic algorithm in population diversity preservation," in *2018 International Conference on Industrial Enterprise and System Engineering (IcoIESE 2018)*. Atlantis Press, 2019, pp. 122–128.

[32] S. Sivanandam, S. Deepa, S. Sivanandam, and S. Deepa, *Genetic Algorithms*. New York, NY, USA: Springer, 2008.

[33] E. P. dos Santos Amorim, C. R. Xavier, R. S. Campos, and R. W. dos Santos, "Comparison between genetic algorithms and differential evolution for solving the history matching problem," in *Computational Science and Its Applications–ICCSA 2012: 12th International Conference, Salvador de Bahia, Brazil, June 18-21, 2012, Proceedings, Part I 12*. New York, NY, USA: Springer, 2012, pp. 635–648.

[34] J. B. Tsui, *Digital techniques for wideband receivers*. SciTech Publishing, 2004, vol. 2, pp. 111–154.

[35] K. R. Rao, D. N. Kim, and J. J. Hwang, *Fast Fourier transform: algorithms and applications*. New York, NY, USA: Springer, 2010, vol. 32, pp. 31, 32.

[36] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965.

[37] H. Jeon, Y. Jung, S. Lee, and Y. Jung, "Area-Efficient Short-Time Fourier Transform Processor for Time–Frequency Analysis of Non-Stationary Signals," *Applied Sciences*, vol. 10, no. 20, p. 7208, 2020.

[38] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*. New York, NY, USA: Springer, 2013 [Online]. Available: https://faculty.marshall.usc.edu/gareth-james/ISL/

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California