UTRECHT UNIVERSITY

COMPUTING SCIENCE

MASTER'S THESIS

A Geometric Approach to Time Delay Estimation for Acoustic Localization of Primates

---

Author:
Siebe VUIJST

Student number:
6810799

Supervisor:
Dr. Frank STAALS

Second examiner:
Dr. Maarten LÖFFLER

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Faculty of Science
Department of Information and Computing Sciences

June 10, 2025

Abstract

Due to the threatened status of multiple howler monkey species, it is important to monitor them for effective conservation. Passive methods such as acoustic localization are becoming increasingly popular for this purpose. This involves recording vocalizations by an array of microphones that are spread over the habitat of the animal and estimating differences in arrival times to triangulate its position. The quality of these time differences therefore largely determines the quality of the position estimate. However, the signal of interest may be significantly obscured by background noise, thereby complicating the accurate estimation of those differences. The state-of-the-art cross-correlation method fails to overcome this difficulty, which calls for more robust methods. This thesis investigates whether techniques from computational geometry may be more successful in this, with the ultimate goal of improving the position estimates. We model the problem of shape matching under one-dimensional translations and propose (multiple variations of) exact or approximation methods for three geometric distance measures: the Hausdorff distance, the Fréchet distance and the Earth Mover's Distance. Experimental results demonstrate that the cross-correlation method is still significantly more robust than our proposed methods. The localization of eight roars captured from the field shows more promising results as the performance of the Hausdorff distance for point sets does not show a statistically significant difference, whereas the other methods lag behind and perform significantly worse. Simulations reveal the limitations and the impact of the microphone geometry on those, which confirm that the largest bottleneck is the scalability.

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisor Frank Staals for his guidance throughout the project and providing invaluable feedback. He always took his time to discuss anything in great detail with me. I also want to thank Yannick Wiegers for introducing this research topic which has been very exciting to work on, for sharing the howler monkey data and other scripts, and for his support during the project. I want to thank Maarten Löffler for being my second examiner. Finally, I want to thank my friends, who made the study an enjoyable experience, and my family for their continued support.

Contents

Chapter 1

Introduction

In the fields of ecology, behavior biology, and conservation biology, it is important to survey and monitor the population and behavior of animals. A common and simple method to achieve this is direct observation of the animals in the field. However, this may not always be practically feasible due to a shortage of resources to observe for long periods of time, or the animals being cryptic, elusive, nocturnal, or living in visually occluded and very large habitats. Another disadvantage of this method is that the presence of human observers can inadvertently affect the observing behavior [53]. An alternative, controversial method is capturing the animals and attaching a tracking device (e.g., GPS) to each individual, but this can be a stressful experience for them that could consequently affect their behavior and survival [79]. Furthermore, these tracking devices can be expensive and are limited by power capacity, requiring frequent change of batteries. Hence, there is a growing interest for cheap, non-invasive and passive methods that can be employed for longer periods of time.

Vocal animals reveal their presence by the sound they produce. Acoustic sensors, which are audio recording devices for recording environmental sound, are becoming increasingly popular for monitoring these animals, including birds, amphibians, bats, and other mammals. With the recent advances in technology, these devices are becoming more practical and affordable. In large-scale wildlife surveys, they have proven to be cost-effective and their performance often exceeds that of human observers [27]. Not only do vocalizations provide information about the presence of the animals, but also about their specific location since sound travels in a predictable way. The recorders can therefore be used to localize the animal in space too. This specific application is known as acoustic localization. Acoustic localization (sometimes referred to as acoustic multilateration) is the process of estimating the location of a sound source by using recordings of its produced sound captured by an array of time-synchronized microphones. See Figure 1.1 for an illustration. It received much attention in marine sciences for studying the behavior and ecology of aquatic animals [65, 67], which are challenging to directly observe, but the use of this method in terrestrial wildlife has thus far been less explored. Likely due to it being more difficult as sound propagates better through water than air. The environment is also often more cluttered and has varying conditions that both affect the acoustics and thereby the detection with sensors [28]. Besides localizing the animals themselves, this technique also shows promising results in species conservation by localizing poachers via gunshot sounds [77] and illegal logging via chainsaw sounds [9].

Among the highly vocal animals, the howler monkeys (genus *Alouatta*) stand out as their loud calls can be heard from up to 4.8 kilometers away through dense forest. They are the loudest terrestrial animals known in the world and outperform all other animals in both call duration and amplitude per body size [26]. The loud calls can have multiple functions, for example avoiding predation, facilitating group cohesion, attracting females and competing with other males or other groups over

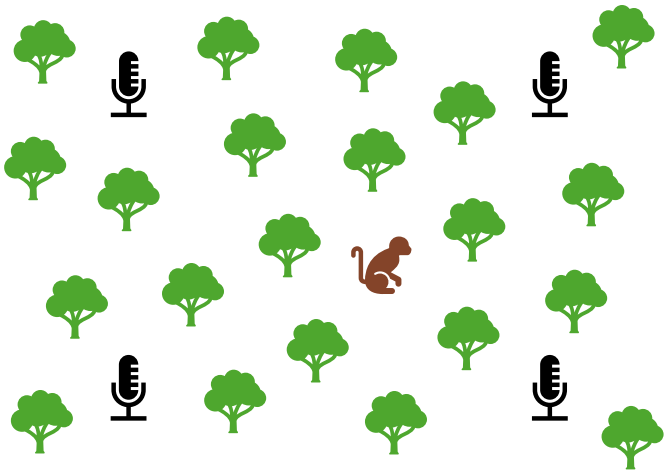


FIGURE 1.1: Setup of an array of four microphones in the habitat of an animal.

resources such as food or space [48]. These animals are native to the forests of Central and South America, where they live in tall rainforest trees in groups of 4 to 19 members. They travel from tree to tree to search for food and are most active during the day. The group of howler monkeys consists of 15 different species that have been recognized. However, according to the Red List of Threatened Species [42] by the International Union for Conservation of Nature (IUCN), the conservation status of this genus is concerning as two species are marked near threatened, five are marked vulnerable and two are marked endangered. The main threats they are currently facing are deforestation and hunting. Due to their (near) threatened status, it is crucial to monitor the populations and take proper measures when necessary to protect them from further extinction. As the loud calls of these species are audible at large distances, acoustic localization could be a promising tool to achieve this goal.

Acoustic localization involves computing the *time difference of arrivals* (TDOAs) of vocalizations recorded by pairs of microphones and use these in a triangulation process to determine the position. State-of-the-art methods for computing the TDOAs are cross-correlation of the representations between the two sound signals, where one representation is slided over the other in time and a correlation coefficient is calculated at each time offset. The peak correlation coefficient then corresponds to the best overlap of the two signals. However, the recordings of the howler monkey vocalizations often have a low signal-to-noise ratio, meaning that the signal is obscured by (background) noise and is therefore difficult to distinguish. The cross-correlation methods are inherently sensitive to noise, which currently results in imprecise TDOA measurements. For accurate localization of the animals, these measurements need to be very precise. Related works on the acoustic localization of primates [25, 66] show that cross-correlation is insufficient to accurately estimate those due to the low signal-to-noise ratios, which is why researchers have to rely on error-prone and time-intensive manual estimation instead. For localization at large scales, this is clearly impractical. To this end, we have to resort to more robust methods. One understudied approach would be to consider the sound signals as geometric entities that form shapes in space. We could then potentially try to match the shapes of two sound signals using methods from computational geometry.

1.1 Research Questions

Therefore, the central research question that we aim to answer in this thesis is:

Does taking a geometric approach for estimating the time difference of arrival between two sound signals improve the localization of howler monkeys?

To answer this question, we have to answer a couple of sub-questions:

- *How to model the problem in a geometric fashion (i.e., what is our geometric input and our desired output)?*

  If we model sound as a function of its intensity over time, the graph of this function represents a shape in space. Given two of those approximately equal shapes, where one is possibly shifted in time with respect to the other, we would like to determine the precise TDOA. We need to formally specify this problem. After we have done that, the next question that arises is:

- *What methods can be used to solve the problem (i.e., how to compute the TDOA between two geometric input representations)?*

  If we have some sort of function that given two shapes returns a quantitative measure of their similarity (also called a similarity measure), we could algorithmically try to minimize this function over all possible shifts in time. After development of such algorithmic techniques, they need to be evaluated where we aim to answer each of the following questions:

- *How accurate are the estimated TDOAs using these methods (i.e., is the method robust enough against noise)?*

- *How accurate are the derived positions (taking into account the errors made on the TDOAs)?*

- *How do the obtained results specifically compare to the state-of-the-art cross-correlation methods?*

1.2 Our Contributions

The contributions in this thesis are twofold. We contribute to the area of computational geometry by studying the problem of shape matching under one-dimensional translations, which has received less attention thus far in the literature, and developing exact or approximation algorithms that solve this problem with respect to three well-known geometric distance measures. The algorithms are moreover implemented and made publicly available [75], enabling the use of them in a variety of applications. The application considered in this thesis is TDOA estimation, which is a crucial part of the localization process. With this, we aim to provide more advanced tools that help in the accurate monitoring of howler monkeys, and potentially primates or vocal animals in general, which contributes to effective wildlife conservation.

1.3 Outline

This thesis is structured as follows. Chapter 2 provides the necessary background information to understand the relevant concepts and describes related and/or previous work. In Chapter 3, we formally study the problem that is related to TDOA

estimation and propose solutions for this. Chapter 4 describes the methodology of this research. The experimental results are presented and discussed in Chapter 5. Finally, a conclusion based on the experimental results is made in Chapter 6, together with possible directions for future research.

Chapter 2

Background

In this chapter, we first introduce core concepts related to acoustic localization, with a focus on hyperbolic methods that require TDOAs to estimate the position. We take a look at previous work that has been done on investigating the call structure of the Guianan red howler monkey, which are the primates of interest in this thesis and in particular their loud calls, and also briefly summarize results obtained from acoustic localization studies on other primates. Finally, we delve into previous work within the topic of shape matching, which is a subarea of computational geometry that concerns our problem, for three geometric distance measures. We provide a literature review of algorithms for computing a distance statically and under transformation, where we are especially interested in the set of transformations that is restricted to translations only. To the best of our knowledge, little to no research has been done before on these distance measures applied to sound signals (or signals in general).

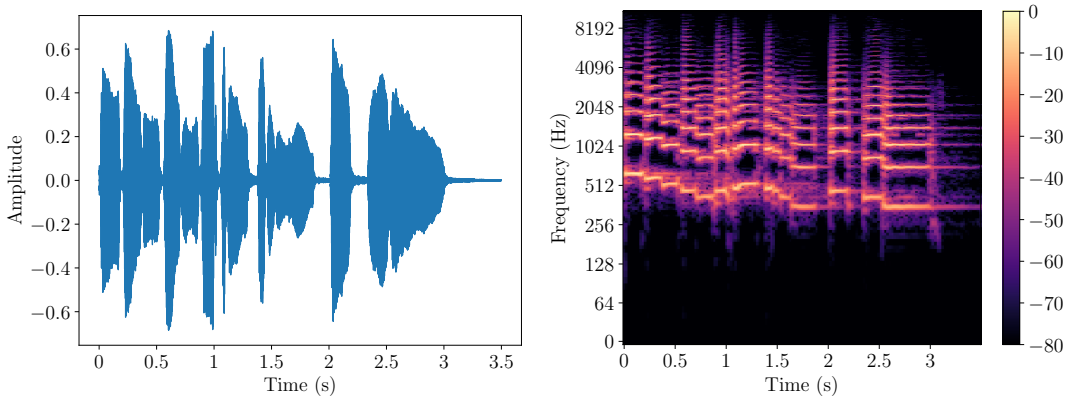
2.1 Acoustic Localization

2.1.1 Sound Propagation

Sound is produced by vibrations which alternately compress and decompress the medium, creating pressure waves that radiate outward from the sound source. A proper understanding of properties of sound is required in order to perform an acoustic localization study, which include the amplitude, frequency and wavelength. Amplitude is proportional to perceived loudness and commonly measured in decibels (dB). Frequency is the number of waves per time unit and is measured in hertz (Hz). It is perceived as pitch with a higher frequency corresponding to a higher pitch and a lower frequency to a lower pitch. The wavelength is the distance between waves and is inversely proportional to the frequency. When a sound is produced, the sound waves move through the medium (e.g., air or water) and propagate from the sound source as a sphere of increasing diameter. As the sound gets farther from the source, the amplitude decreases. This process is called attenuation. Sounds with higher frequency attenuate faster than sounds with lower frequency, which means that they reach a smaller area and become less likely of being received at larger distances.

Note that attenuation affects the detection of a signal produced by vocalizing animals. Animals that produce louder vocalizations can be detected at larger distances as the amplitude of the signal is larger. However, if the animals produce equally loud vocalizations at different frequencies, the signal with the lower frequency will be detectable at a larger distance than the signal with the higher frequency.

Properties of the medium also affect the sound propagation as they determine the speed of sound, which is the speed at which the pressure waves travel. Sound waves travel almost five times faster in water than in air due to its higher density. In air, it is largely determined by factors such as temperature, humidity and pressure.



(A) Waveform (B) Spectrogram

FIGURE 2.1: Visualizations of digital audio data (example from a trumpet).

2.1.2 Digital Sound

When a sound wave arrives at a microphone, it is converted to an electric signal. During digital recording, the amplitude of the electric signal is sampled at a given sample rate (typically 44.1 kHz) and bit-depth (typically 16-bit). This records the signal in the time-domain, visualized as a waveform where the x-axis is the time and y-axis is the amplitude (see Figure 2.1a). In this domain, we can observe how the signal changes over time. The sample rate affects the frequency resolution, with the Nyquist–Shannon sampling theorem stating that the sampling rate must be at least twice as high as the highest frequency of interest. The bit-depth affects the amplitude resolution, so a higher bit-depth corresponds to a better approximation for the digital representation of the original sound wave.

To recover the frequency information and shift our focus to the frequency-domain representation, which is often more insightful than its time-domain representation, a mathematical technique called Fourier Transform is applied to the data. As the digital audio data is inherently discrete, this specifically requires the Discrete Fourier Transform (DFT). The main idea behind this technique is to decompose the signal into a sum of sinusoids of different frequencies. A sinusoid (also called a sine wave) is defined as:

$$y(t) = A\sin(2\pi ft + \phi)$$

where A denotes the amplitude, f denotes the frequency and ϕ denotes the phase. Let x be a discrete signal represented as an array of N samples where $x[n]$ is the amplitude value at the n-th sample. Then the frequency representation $X[k]$ is computed by:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}kn}$$

The output X is an array of complex numbers, consisting of real and imaginary components. This holds information about the amplitude and phase of a sinusoidal component for each frequency component. The array is again of size N. Let f_s be the sampling rate of the original time-domain signal, then the frequency f_k corresponding to the k-th bin is calculated as follows:

$$f_k = \frac{k \cdot f_s}{N}$$

The amplitude of a frequency component is obtained by taking the magnitude of the complex number $X[k]$. Since the input signal is real-valued, the spectrum of the positive and negative frequencies is symmetric: the first half of the DFT output ($k = 0$ to $N/2$) represents the positive frequencies and the second half ($k = N/2 + 1$ to $N - 1$) is the complex conjugate of the first half. This means that only the first half needs to be considered for obtaining amplitudes of unique frequencies. An efficient algorithm that is often used to compute the DFT is Fast Fourier Transform (FFT) which computes the representation in $O(N \log N)$ time.

While applying DFT over the entire signal provides insights of the occurring frequencies, it does not provide information about how the frequencies change over time. This requires the Short-Time Fourier Transform (STFT). The main idea behind this technique is to take segments of the signal and compute the DFT for each segment. It is mathematically defined as:

$$X[m,k] = \sum_{n=0}^{M-1} x[n + mH] \cdot w[n] \cdot e^{-j\frac{2\pi}{M}kn}$$

where M is the window size, which defines the number of samples per segment, H is the hop size, which determines the amount of overlap between consecutive segments, and $w[n]$ is the window function (e.g., Hamming, Hanning or Gaussian) which is also of size M. The purpose of applying this window function to the segment is to reduce the effect of leakage that occurs due to discontinuities at the boundaries of the segment. These parameters naturally affect the quality of the frequency-domain representation. For the window size, there is a trade-off between the time and frequency resolution. A shorter window size allows for a higher time resolution, which captures rapid changes in the signal, but a lower frequency resolution, which makes it harder to distinguish closely spaced frequency components. Conversely, a larger window size allows for a higher frequency resolution, which helps to identify smaller frequency differences, but a lower time resolution, so identifying the rapid changes in the signal becomes more difficult. It is therefore crucial to find an optimal balance between the two. The overlap rate helps to reduce the loss of information at the edges of the windows and with that providing a smoother representation. While a higher overlap rate generally results in a smoother representation, it comes with a higher computational cost. The time-frequency domain representation can then be visualized using a spectrogram, where the x-axis is the time, the y-axis is the frequency and the color intensity is the amplitude (see Figure 2.1b). It is important to note that the waveform and spectrogram are just different ways to visualize the same data.

2.1.3 Signal-to-Noise Ratio

The signal-to-noise ratio (SNR) is a measure that compares the level of the desired signal to the level of the background noise. Formally, it is defined as the ratio of the power of the signal to the power of the background noise:

$$\text{SNR} = \frac{P_{signal}}{P_{noise}}$$

where P denotes the average power. There are multiple ways for computing the average power. One concrete example is: Suppose we have a discrete signal $x[t]$ and a noise $n[t]$, both consisting of N samples. We can calculate the average power of the signal P_{signal} as follows:

$$P_{signal} = \frac{1}{N} \sum_{t=1}^{N} x[t]^2$$

Analogously for the average power of the noise P_{noise}. The SNR that results from these powers can take a wide range of values, which makes it difficult to intuitively interpret. We can convert this ratio to decibels (dB), which is often used in signal processing, as follows:

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \left(\frac{P_{signal}}{P_{noise}} \right)$$

A positive SNR implies a good signal quality where the signal power is stronger than the noise. On the other hand, a negative SNR implies a poor quality where the noise dominates the signal. This has implications on the performance and quality of signal processing algorithms, including the estimation of the time difference of arrival.

2.1.4 Time Difference of Arrival

Consider a deployed array of microphones in an area and some source produces a sound near (or within) the array. The arrival of the sound at each microphone is then delayed by a certain amount of time. If the distance from the source to the microphone is d meters and the speed of sound in that environment is c meters per second, the time of arrival (TOA) in seconds is calculated as follows:

$$TOA = \frac{d}{c}$$

The microphones of an array are positioned at different locations. This means that the sound travels a different distance to reach each microphone in general and consequently arrives at different times. The difference between the arrival times of a sound between two microphones is called the time difference of arrival (TDOA). For a sound captured by two microphones with time of arrivals TOA_1 and TOA_2, the TDOA is calculated by:

$$TDOA_{12} = TOA_2 - TOA_1$$

which tells us how much later the sound arrived at the second microphone compared to the first microphone.

In a typical application, the TDOA tends to be in the order of several milliseconds. To accurately measure this difference, proper synchronization of the recorders is required. Synchronization is the process of temporally aligning recordings from multiple microphones. Often times, periodic re-synchronization is necessary as recorders will eventually fall out of synchronization, even if they start recording simultaneously. This phenomenon is known as *drift* and is caused by slight differences in true sampling rates of the recording hardware. There are many methods to synchronize the recorders that have been reviewed by Rhinehart *et al.* [62]. Cable synchronization connects the microphones to a central multichannel recorder which records the captured sounds simultaneously, so synchronization is done during recording. Acoustic

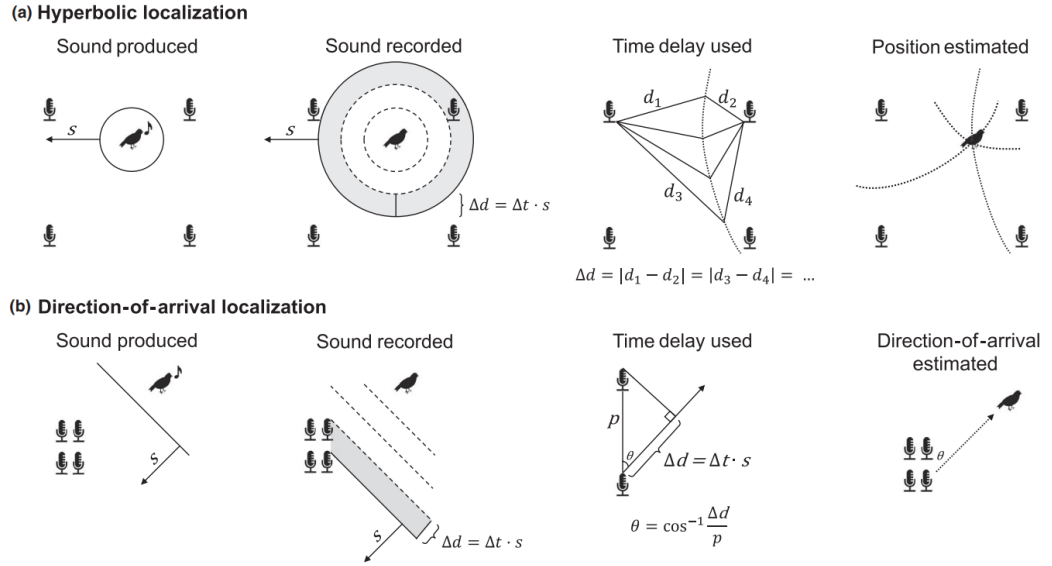


FIGURE 2.2: Wildlife localization approaches (from Rhinehart *et al.* [62]).

synchronization involves playing back an artificial sound from a known location and subsequently computing the expected delay in arrival time at each microphone. The recordings are then aligned to these delays. These two methods often work for arrays that cover a relatively small area, but for larger areas, a feasible alternative would be GPS synchronization. In this method, a GPS receiver is attached to each microphone and the received GPS timestamps are then used to align the recordings, which can be done during or after the recording process. If using GPS in a specific environment could be unreliable for some reason, another option would be to synchronize by connecting each recorder to a shared wireless network.

2.1.5 Hyperbolic Localization

Reviewed by Rhinehart *et al.* [62], localization approaches are divided into two broad categories: hyperbolic and direction of arrival (DOA; see Figure 2.2). Recall that, when a sound is emitted, the sound waves radiate from the source location as a sphere. This is the near field assumption that hyperbolic methods make, which requires the array of microphones to be widely spaced where the distance from the sound source to the microphones must be around the same order of magnitude as the distance between the microphones [50]. These methods also often require the explicit calculation of TDOAs, which is why they are referred to as TDOA localization algorithms. Conversely, if the sound originates from a distant source and arrives at a set of at least four closely spaced microphones, the curved edge of the arriving sound can be approximated as a straight line. This is the far field assumption that DOA methods make. In this case, a single set of microphones is able to derive the direction of arrival of a sound. The location can then be determined by intersecting the direction of arrival estimates obtained from at least two sets. There are other methods of localization that do not precisely fall into one of the two mentioned categories (e.g., time-of-arrival and energy-based [43]), but they are less suitable for wildlife localization due to the lack of essential information or higher inaccuracies in such a setup [62].

For the hyperbolic methods, four microphones are required to unambiguously determine the position on the plane and five microphones for positioning in the three-dimensional space [64]. However, more microphones generally results in a higher localization accuracy as averaging the results reduces the influence of errors made by any recorder.

The acoustic recorders often record the sound from the environment for the duration of several minutes, which contain segments of interesting parts (i.e., vocalizations of animals) and background noise. This data must be either manually or automatically analyzed to identify the segments of interest (e.g., recent developments in deep learning allow for very accurate automatic identification [68]). After identifying the sounds that are desired to be localized from the recordings, the time difference of arrival must be derived for pairs of recordings. A popular method for directly calculating those is spectrogram (or waveform) cross-correlation, which was first introduced by Clark *et al.* [22]. This technique involves overlaying the two representations and sliding one over the other incrementally in time and calculating a correlation coefficient at each time offset. The peak correlation coefficient then corresponds to the time offset between the two representations. For two signals $X[t, f]$ and $Y[t, f]$ where t and f represent the time and frequency respectively in the spectrogram representation, the cross-correlation $R_{XY}(\tau)$ measures the similarity between X and Y as a function over the time lag τ and is calculated as follows:

$$R_{XY}(\tau) = \sum_{t} \sum_{f} X[t, f] \cdot Y[t + \tau, f]$$

The waveform cross-correlation is similarly calculated (without the additional frequency component) and is often a more accurate method in general due to spectrograms having imperfect temporal resolution, but automatically identifying the sounds of interest in a waveform requires higher signal-to-noise ratios [78]. It is also possible to manually obtain the TDOA by visual inspection, but this is time-intensive and error-prone.

Using the obtained TDOAs, position estimation algorithms are used to finally determine the location of the sound source. This is the two-stage approach, where we first explicitly calculate the TDOAs and then use them as input to the algorithm for estimating the position. There are also algorithms following a one-stage approach, which implicitly use the TDOA information without explicitly calculating the TDOAs first. Consider the time difference of arrival Δt between two microphones in seconds. If the speed of sound is c meters per second, then we can calculate a distance Δd as follows:

$$\Delta d = c \cdot \Delta t$$

This is the difference of distance of the sound source between the two microphones. This defines a set of potential locations of the origin of the source, which forms a contour that has the shape of a hyperbola in the plane and hyperboloid in three-dimensional space (hence the term "hyperbolic" in hyperbolic localization). Each pair of microphone defines such a contour and the intersection of those should ideally give the exact location (see Figure 2.2(a)). If there is no such perfect intersection, which is very likely in practice due to inaccuracies in the measurements, the algorithms may estimate the point that, for example, minimizes the sum of the squared distances to the contours (or some other optimization criterion). The (Euclidean) distance between the true and estimated position is called the position estimation error. Rhinehart *et al.* [62] reviewed the main causes of the position estimation error:

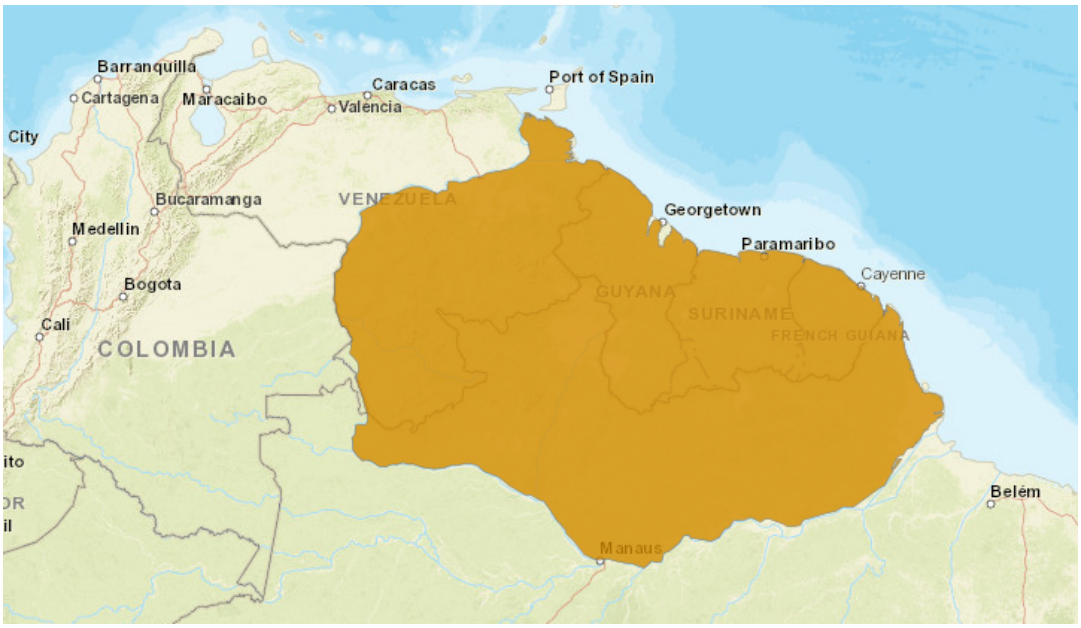


FIGURE 2.3: Distribution map of the Guianan red howler monkey (from IUCN [42]).

- Using fewer microphones than the recommended number, which creates some areas of localization ambiguity (further explained in [64]).
- No tight and frequent (re-)synchronization, which results in inaccurate TDOA measurements.
- Inaccurate estimation of speed of sound, which depends on the temperature, wind and humidity conditions that also require precise measurements at any time.
- Improper placement of the microphones.
- Inaccurate measurement of the microphone locations.
- The location of the sound source with respect to the array: localization inside the convex hull of the array is more accurate than outside and closer to the center of the array is more accurate than closer to the edges.
- Errors in calculating the TDOAs caused by external factors such as background noise, attenuation of the sound where the amplitude decreases farther from the source, and reverberation due to reflecting sound in the environment, which is more common in forest habitats than in open fields. These factors distort the signal and affect the signal-to-noise (SNR) ratio.

2.2 Guianan Red Howler

The red howler is the most common howler monkey species with five subspecies being recognized. One of them is the Guianan red howler monkey (*Alouatta macconnelli*), who can be found in Guiana, French Guiana, Trinidad, Suriname, east of the Orinoco River in Venezuela, and north of the Amazon River in Brazil (see Figure 2.3). Particularly in areas such as rain forests near sea levels, gallery forests and dry

deciduous forests. The estimated population density in French Guiana is around 19 individuals per km^2 where they live in groups of 5 members on average (and range from 2-10). According to the most recent IUCN Red List of Threatened Species assessment in 2021 [42], the species is listed as "Least Concern" due to its widespread distribution and lack of any major threats that could result in a significant population decline. However, they are still hunted locally and in southern parts of their range deforestation occurs due to ranching.

2.2.1 Acoustic Structure

The acoustic structure of the loud calls and the 24-h emission behavior for this specific species have been studied extensively by Drubbel and Gautier [73] in French Guiana. These loud calls are described as low-pitched noisy hoarse sounds with an upper frequency limit of around 3 kHz and roughly divided into two classes based on their duration: long (more than 60 seconds) and short (less than 40 seconds) calls (which have similar carrying distances). The long calls have a mean duration of 3 minutes and 28 seconds with a maximum of 10 minutes and standard deviation of 109 seconds (over 603 samples). It consists of three successive phases: the introduction (on average 60 seconds), the climax (on average 120 seconds) and the coda (on average 1 second). At the end of the call, after a period of silence for 2 seconds, one or two blowing sounds (on average 3.3 seconds) are heard with again a period of silence in between (on average 7.4 seconds). These blowing sounds have a relatively low amplitude with a low carrying distance, which are likely produced by emptying of the air sacs. During the introduction phase, the amplitude gradually increases and reaches a maximum at the start of the climax phase. Then, during the coda phase, the amplitude gradually decreases again and reaches a minimum at the start of the blowing sounds. The maximum bandwidth of the loud calls is between 125 and 3128 Hz where two clusters of frequency bands were observed in the introduction and climax phases: 310-1100 and 1310-2900 Hz. Each cluster includes two dominant frequency bands of around a 250-Hz range. The short calls have a mean duration of 11 seconds with a standard deviation of 11.2 seconds (over 122 samples). The analysis of this call type is limited as these calls were mainly emitted during daytime where the spectral energy hardly contrasted with the background noise. In contrast, the long calls were mostly heard during the night, with a peak around dawn. At both day and night, the number of calls are on average 5.2. The total call duration was 15 minutes during night and 8 minutes during daytime.

The previous study, which was conducted over three decades ago, was constrained in software and hardware equipment that limited the analysis of both the spectrograms and number of calls. Recently, Do Nascimento *et al.* [30] performed a study on the differences of the call structure over the entire diurnal cycle in the Viruá National Park, Roraima, Brazil. They only consider the long calls and ignore the short calls (which they refer to as barks) in their analysis. A statistical test confirmed that calls at night are not only more common than calls during the day, but also on average shorter in duration (258 vs 327 seconds). The mean and median frequencies are always around 900-1000 Hz and the dominant frequency (frequency with the highest amplitude) is around 700 Hz. Moreover, the calls during the day have 5% lower frequencies and are harsher as measured by a lower (1.16 vs 2.01) harmonic-to-noise ratio (which is a measure of deterministic chaos). This leads to calls that sound more intimidating.

2.2.2 Related Work

While there has been done extensive research on the call structure of the Guianan red howler monkey, acoustic localization of these species remains an understudied topic. However, there has been done related research on acoustic localization of other primates (i.e., Eastern chimpanzee and Bornean orangutan), which we summarize below.

Fairly recently, Crunchant *et al.* [25] performed an acoustic localization study for localizing the Eastern chimpanzee (*Pan troglodytes schweinfurthii*) in western Tanzania. They deployed an array of four GPS time-synchronized acoustic sensors in a mountainous environment with heterogeneous vegetation. The sensors were placed around 500 meters from each other that cover an area of nearly 2 km^2. The collected sound samples were visualized as spectrograms in the Raven software [61] and the TDOAs were manually estimated due to the low signal-to-noise ratio of the chimpanzee calls. The position was subsequently estimated in the Sound Finder software [78] that uses the least-squares solution which was developed for global positioning systems [12]. From the playback experiments, they show a mean localization error of 27 meter and a standard deviation of 21.8 meter. The localization error increased with higher temperatures and surprisingly with lower wind speeds. The localization also appeared to be more prone to error in open vegetation than closed vegetation.

Spillmann *et al.* [66] tested the utility of an Acoustic Localization System (ALS) for localizing the Bornean orangutan (*Pongo pygmaeus wurmbii*) in Central Kalimantan at the Tuana field site, a dense peat swamp forest. They deployed an array of 20 GPS time-synchronized acoustic sensors that were placed in a lattice at 500 meter intervals encompassing a grid of 3 km^2. The study aimed to validate this system in two main steps. The first step concerned determining whether it is able to pick up all long calls in the area and the second step was about checking whether the system provides accurate localizations. For this, observers collected the GPS locations of 89 long calls to compare against the estimated position. For identifying the long calls, they trained a recognition model using the Song Scope software (no longer supported as of 2016), which was able to find 99% of the long calls. Analysis also showed that the long calls were often still picked up by a recorder at 700 meter distance, which means the array covered an area of around 9 km^2 in total. For estimating the time of arrival differences, they used the spectrogram cross-correlation that is provided in the Raven software. A band-pass filter from 200 to 1000 Hz is additionally applied to remove background noise. They had to manually inspect the automatically generated correlation functions due to the low quality of the signal of interest, which resulted in imprecise time of arrival differences. Similar to the previously mentioned work, they use these as input to the Sound Finder software to estimate the position. Triangulation of 66 long calls that occurred within the hull of the array resulted in a mean error of 58 meter and standard error of mean of 7.2 meter. Considering the error made by variable animal-human distances and the measured coordinates of the GPS (approximately 8-12 meter), they found the results reasonable compared to the inter-individual distances between the animals. Long calls outside the grid furthermore showed much larger triangulation errors, especially if the distance from the long call to the grid is more than 200 meter.

2.3 Shape Matching

The term shape is used for describing a geometrical pattern that consists of a set of points, curves, surfaces, and other geometric entities. Shape matching is important in

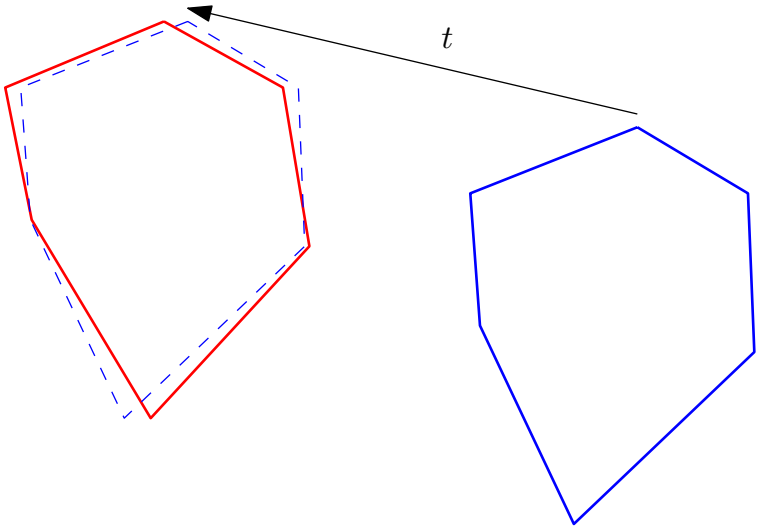


FIGURE 2.4: Matching polygons under translations.

various application areas, including computer vision, pattern recognition, computer-aided design and robotics. It involves transforming the shape (e.g., by translation, rotation or scaling) and measuring the resemblence with another shape using some similarity measure [7, 71]. One well-known application in multimedia retrieval is the retrieval of shapes from a database that are similar to a query shape, where the similarity is invariant of transformations (see Figure 2.4 for an illustration). The matching problem has been studied in various forms which have been classified accordingly by Veltkamp and Hagedoorn [71]. Informally, given two shapes and a distance function:

Computation compute the distance between the two shapes.

Decision (static) decide whether the distance between the two shapes is smaller than a given threshold.

Decision (dynamic) decide whether there exists a transformation such that the distance between the transformed shape and other shape is smaller than a given threshold.

Optimization find the transformation that minimizes the distance between the transformed shape and the other shape.

Ideally, we can efficiently compute an exact solution to the optimization problem, but due to its complexity and the high computational cost that comes with this, it is not always practically feasible. In that case, we have to rely on approximations of the solutions instead. This means that we obtain a solution of a transformation where the distance between the transformed shape and other shape is within a certain factor c from the minimum distance. We call this a c-approximation.

There are multiple ways of approaching the shape matching problem, but in this thesis we specifically focus on methods from computational geometry. The distance function is in that case of geometric nature. We will be looking at three such measures that are relatively popular in the literature, namely the Hausdorff distance, Fréchet distance and Earth Mover's Distance.

2.3.1 Metric Property

It is often desired that the similarity measure is a *metric*. Let S be a set of objects (e.g., collection of shapes) and $d : S \times S \to \mathbb{R}$ be some function. Then d is a metric if and only if it satisfies the following properties for all $x, y, z \in S$:

Non-negative $d(x, y) \geq 0$ and $d(x, y) = 0$ if and only if $x = y$

Symmetry $d(x, y) = d(y, x)$

Triangle inequality $d(x, y) \leq d(x, z) + d(z, y)$

If d is a metric, then the pair (S, d) is called a *metric space*. If we replace the first condition such that two distinct elements can be zero, i.e., $d(x, x) = 0$, then d is called a *pseudo-metric*.

2.3.2 Hausdorff Distance

The Hausdorff distance is one of the most widely used distance measures. Informally, it is the maximum of all the distances from a point in one set to the closest point in the other set. Below we provide a formal definition.

Definition

For two compact subsets $A, B \subseteq \mathbb{R}^d$, we first define the *directed* Hausdorff distance $h(A, B)$ as the maximum of distances between each point in A to its nearest neighbour in B:

$$h(A, B) = \max_{a \in A} \min_{b \in B} d(a, b)$$

where d is the underlying distance (e.g., Euclidean). Note that this function is not symmetric (i.e., $h(A, B) \neq h(B, A)$). The *undirected* Hausdorff distance $H(A, B)$ is then the maximum over the directed distances:

$$H(A, B) = \max\{h(A, B), h(B, A)\}$$

When we are referring to the Hausdorff distance, we typically mean the undirected distance.

Computation

For polygons that consist of m and n vertices, the Hausdorff distance can be computed in $O((m + n) \log(m + n))$ time using Voronoi diagrams [4]. This method also works for point sets in the plane and sets of non-intersecting line segments in the plane.

For point sets of size m and n (in any dimension), a straightforward approach to compute their (directed) Hausdorff distance is brute-force, where we have a pair of nested loops iterating over the two sets while keeping track of the minimum distance in the inner loop and the maximum distance in the outer loop. This clearly takes $O(mn)$ time. In theory, this is the best possible algorithm known in terms of the worst-case complexity, but there has been done research to improve the runtime using heuristics and by exploiting the structure of the input. The inner loop of the brute-force algorithm is an exhaustive nearest neighbor search, but an exhaustive search may actually not be necessary: If we compute a distance between a pair of points that is lower than the minimum Hausdorff distance known thus far, we may

skip to the next iteration of the outer loop as further computation will not affect the Hausdorff distance. This gives a linear best-case time complexity for computing the directed Hausdorff distance. In combination with an effective sampling strategy, this can significantly reduce the runtime. An algorithm exploiting this early break condition with random sampling was presented by Taha and Hanbury [69].

Translation-invariant

For polygons that consist of m and n vertices, there is an algorithm known specifically for translations along one fixed direction only that runs in $O(mn\log(mn)\log^*(mn))$ time by Alt *et al.* [4]. They claim that this algorithm also works on more general structures like polygonal chains. For other shapes, the focus has been mainly on translations in arbitrary directions.

For point sets in one-dimension of size m and n, Huttenlocher and Kedem [40] presented one of the first algorithms for minimizing the Hausdorff distance under translations. They analyze the structure of the cost as a function of the translation that describes the Hausdorff distance and exploit these properties to design an algorithm that runs in $O(mn\log(mn))$ time.

For point sets in the plane, the Hausdorff distance under translation can be computed in $O(mn\log^2(mn))$ time when the underlying metric is L_1 or L_∞ [21]. This algorithm makes use of segment trees. If the underlying metric is L_2, there is a $O(mn(m+n)\log(mn))$ time algorithm that uses the upper envelopes of Voronoi surfaces [41]. We will describe this method by Huttenlocher *et al.* in more detail.

Huttenlocher *et al.* Given a set $S = \{p_j \mid j = 1,...,n\}$ of points in $\mathbb{R}^d$ and some metric ρ. We denote the Voronoi diagram of S as $Vor(S)$, which is the decomposition of $\mathbb{R}^d$ into "Voronoi cells" $C_1,...,C_n$ such that each cell C_j contains those points of $\mathbb{R}^d$ that are closer to p_j than any other point. Consider the function:

$$d(x) = \min_{p_j \in S} \rho(x, p_j)$$

The graph of this function $\{(x, d(x)) \mid x \in \mathbb{R}^d\}$ is called the Voronoi surface. The surface is at a local minimum when x is coincident with a point $p_j \in S$ and at a local maximum for certain points that lie along the boundary of the Voronoi cells of $Vor(S)$. Let $\{S_i \mid i = 1,...,m\}$ be m point sets, and $n_i = |S_i|$ for $i = 1,...,m$ be the number of points in S_i. Let $n = \sum_{i=1}^{m} n_i$ be the total number of points. The Voronoi surface of a set S_i is denoted as $d_i(x)$. Consider the function:

$$f(x) = \max_{i=1,...,m} d_i(x)$$

The graph of this function is the upper envelope of the m Voronoi surfaces. Note that $f(x)$ is the largest distance from x to each nearest neighbour of S_i.

One application of this upper envelope of Voronoi surfaces is the Hausdorff distance under translation. Let $A = \{a_1,...,a_m\}$ and $B = \{b_1,...,b_n\}$ be point sets. The minimum Hausdorff distance $D(A,B)$ is defined as:

$$D(A,B) = \min_t H(A, B \oplus t)$$

where $B \oplus t$ is the Minkowski sum of B and t and H is the Hausdorff distance as given by the definition above. The key of determining this minimum distance is by finding the value of t that minimizes the upper envelope of the Voronoi surfaces defined by the $m + n$ sets $S_i = a_i \ominus B = \{a_i - b_j \mid b_j \in B\}$ and $S'_j = A \ominus b_j = \{a_i - b_j \mid a_i \in A\}$. For any $a_i \in A$, $b_j \in B$ and translation t, we use the fact that $\rho(a_i, b_j + t) = \rho(a_i - b_j, t)$. We obtain the Voronoi surface of S_i by:

$$d_i(t) = \min_{p \in S_i} \rho(p, t)$$

Similarly, for the set S'_j we obtain the Voronoi surface $d'_j(t)$. We denote by $f(t)$ the upper envelope of the functions $d_i(t)$ and $d'_j(t)$, and then by definition:

$$f(t) = \max\{\max_{a_i \in A} d_i(t), \max_{b_j \in B} d'_j(t)\} = H(A, B \oplus t)$$

Hence, the minimum Hausdorff distance corresponds to the point t that minimizes this upper envelope:

$$D(A, B) = \min_t f(t)$$

The authors show that the local minima of $f(t)$ are on its vertices and show bounds on the number of those. They propose efficient algorithms for computing these vertices in $\mathbb{R}^2$ and $\mathbb{R}^3$, which provides the global minimum required to solve this problem. They also obtained results on line segments in the plane for the L_1 and L_∞ metrics.

The bounds on these have not been improved for over decades. Recent work has shown a lower bound of $(nm)^{1-o(1)}$ for L_1 and L_∞ assuming the Orthogonal Vectors Hypothesis (OVH) and a lower bound of $n^{2-o(1)}$ for L_2 in the imbalanced case of $m = O(1)$ assuming the 3SUM Hypothesis [17]. There is also an approximation algorithm known that runs in $O((m + n)\log(m + n))$ time which uses reference points [3], described in more detail below.

Reference Points. A reference point of a shape is a characteristic point such that similar shapes have reference points that are close to each other. The approach of reference points extends to higher dimensions as well. This framework plays an important role in approximate shape matching. Let C^d be the set of compact subsets of $\mathbb{R}^d$ and let $\mathcal{T}$ be the set of transformations in this space. A mapping $s : C^d \to \mathbb{R}^d$ is called a reference point with respect to $\mathcal{T}$, if the following holds for all $A, B \in C^d$ and $T \in \mathcal{T}$:

Equivariance $s(T(A)) = T(s(A))$.

Lipschitz continuity There exists some constant $c \geq 0$ such that:

$$d(s(A), s(B)) \leq c \cdot H(A, B)$$

where d is the Euclidean distance. We call c the quality of the reference point. If we have such a mapping s with respect to a set of translations $\mathcal{T}$, then the translation of B that approximately minimizes the Hausdorff distance is simply $s(A) - s(B)$. This algorithm provides a $(c + 1)$-approximation. So the approximation guarantee depends on the quality of the reference point.

In the two-dimensional case, it was observed by Alt *et al.* [4] that the point $s(A) = (x_{min}, y_{min})$, where x_{min} and y_{min} are the minimal x and y coordinates of

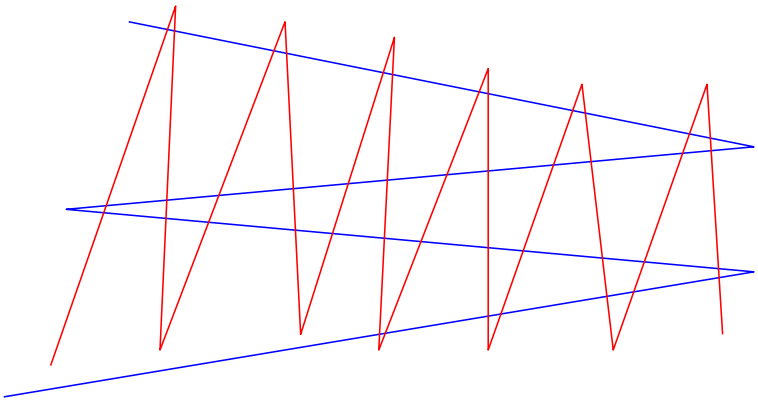


FIGURE 2.5: Two curves with a small Hausdorff distance.

points in A, is a reference point of quality $\sqrt{2}$ for translations. A better reference point was later found by Alt *et al.* [3], namely the Steiner point that has quality $4/\pi \approx 1.27$ which also works on more general transformation sets. It is shown that for any two compact sets A and B and their convex hulls $conv(A)$ and $conv(B)$, we have that $H(conv(A), conv(B)) \leq H(A, B)$. It follows that if we have a reference point for the convex hull of A, then it is also a reference point for A itself. This property allows to focus on convex figures and makes the Steiner point a perfect candidate as reference point (since the definition is quite involved, we refer to the paper for this [3] instead). Although there is evidence that for translations in $d \geq 2$ the quality of any reference point cannot be smaller than $\sqrt{4/3} \approx 1.155$, it is believed that the Steiner point is the optimal reference point.

2.3.3 Fréchet Distance

The Fréchet distance is a measure of similarity between (polygonal) curves. We first provide a formal definition of (polygonal) curves.

Definition (Curves)**.** *A* curve *is a continuous mapping* $f : [a, b] \to \mathbb{R}^d$ *with* $a, b \in \mathbb{R}$ *and* $a < b$*. A* polygonal curve *is a curve* $P : [0, n] \to \mathbb{R}^d$ *with* $n \in \mathbf{N}$*, such that for all* $i \in \{0, ..., n-1\}$ *each* $P|_{[i,i+1]}$ *is affine, i.e.,* $P(i + \lambda) = (1 - \lambda)P(i) + \lambda P(i + 1)$ *for all* $\lambda \in [0, 1]$*. Note that* $P|_{[i,i+1]}$ *means the restricted function of* P *to the domain* $[i, i + 1]$*.*

The Hausdorff distance is considered an appropriate measure in many applications that involve point sets, but for polygonal curves, it is often inadequate. There are examples of curves that have a small Hausdorff distance, but they do not resemble each other at all (see Figure 2.5). The Hausdorff distance only takes into account the sets of points on both curves, thereby ignoring the course of the curves. While this makes the Hausdorff distance easily computable, it may not be a desirable property in some applications. The Fréchet distance is an alternative metric that does not suffer from this issue.

Definition

A popular intuitive definition is as follows: We are given two curves in space. Suppose a man is walking his dog, where he is walking on one curve and the dog on the other curve. They are both allowed to control their speed, but not allowed to go backwards. The Fréchet distance is then the minimal length of a leash that is necessary for both to traverse their paths from start to finish. Below we provide a formal definition.

Continuous. Let $f : [a, a'] \to \mathbb{R}^d$ and $g : [b, b'] \to \mathbb{R}^d$ be curves. Then the Fréchet distance is denoted by $\delta_F(f, g)$ and defined as:

$$\delta_F(f,g) = \inf_{\substack{\alpha:[0,1]\to[a,a'] \\ \beta:[0,1]\to[b,b']}} \max_{t\in[0,1]} d(f(\alpha(t)), g(\beta(t)))$$

where d is the underlying distance (typically Euclidean) and α, β range over continuous and increasing functions with $\alpha(0) = a$, $\alpha(1) = a'$, $\beta(0) = b$ and $\beta(1) = b'$ only.

This definition is known as the classic *continuous* Fréchet distance. A variant is the *discrete* Fréchet distance which intuitively replaces each curve by a sequences of points where at any time step the man and his dog must be at the points of the curves and may jump to the next point. This closely approximates the continuous Fréchet distance and is to some extent easier to compute, but still requires roughly quadratic time assuming the Strong Exponential Time Hypothesis (SETH) [2, 15]. Below we provide a formal definition, following from [44].

Discrete. Given a polygonal curve $P = (p_1, p_2, ..., p_n)$, a k-walk along P partitions the vertices of P into k disjoint nonempty subsets $\{P_i\}_{i=1,2,...,k}$ such that $P_i = (p_{n_{i-1}+1}, ..., p_{n_i})$ and $0 = n_0 < n_1 < ... < n_k = n$. Now given two polygonal curves P and Q of m and n vertices respectively, a paired walk along P and Q is a k-walk along P and a k-walk along Q such that for $1 \leq i \leq k$, either $|P_i| = 1$ or $|Q_i| = 1$ (that is, either P_i or Q_i contains exactly one vertex). The cost of a paired walk $W = \{(P_i, Q_i)\}$ along P and Q is defined as:

$$d_{\mathcal{F}}^W(P,Q) = \max_i \max_{(p,q)\in P_i\times Q_i} d(p,q)$$

The discrete Fréchet distance between two polygonal curves P and Q is then defined as:

$$d_{\mathcal{F}}(P,Q) = \min_W d_{\mathcal{F}}^W(P,Q)$$

Computation

Alt and Godau [6] showed how to compute the continuous Fréchet distance between two polygonal curves. We will describe this method in more detail as the concepts play an important role in computing the translation-invariant version as well. We assume the Euclidean norm, but they also apply to other norms.

Alt and Godau. Let $P : [0, m] \to \mathbb{R}^2$ and $Q : [0, n] \to \mathbb{R}^2$ be two polygonal curves and assume that ε is given and fixed. The *free space* of P and Q is defined as $F_\varepsilon(P, Q) = \{(s, t) \in [0, m] \times [0, n] \mid d(P(s), Q(t)) \leq \varepsilon\}$ (also abbreviated as F_ε). The partition of $[0, m] \times [0, n]$ in points that either belong or not belong to F_ε is referred to as the *free space diagram* $FD_\varepsilon(P, Q)$ (also abbreviated as FD_ε).

Each point $p \in F_\varepsilon$ is called *feasible* or 'white' and each other point $p \in FD_\varepsilon \setminus F_\varepsilon$ is called *infeasible* or 'black'. See Figure 2.6 for an illustration. The free space diagram FD_ε is divided into mn cells $\zeta_{i,j} = [i, i + 1] \times [j, j + 1]$ for $0 \leq i \leq m - 1$ and $0 \leq j \leq n - 1$. We denote by P_i the i-th vertex of P and $\overline{P_i}$ the line segment between P_i and P_{i+1} (analogously for Q). It follows that F_ε is composed of the mn free spaces $F_\varepsilon(\overline{P_i}, \overline{Q_j}) = F_\varepsilon(P, Q) \cap \zeta_{i,j}$ for each pair of line segments $(\overline{P_i}, \overline{Q_j})$ with $i = 0, ..., m - 1$ and $j = 0, ..., n - 1$.

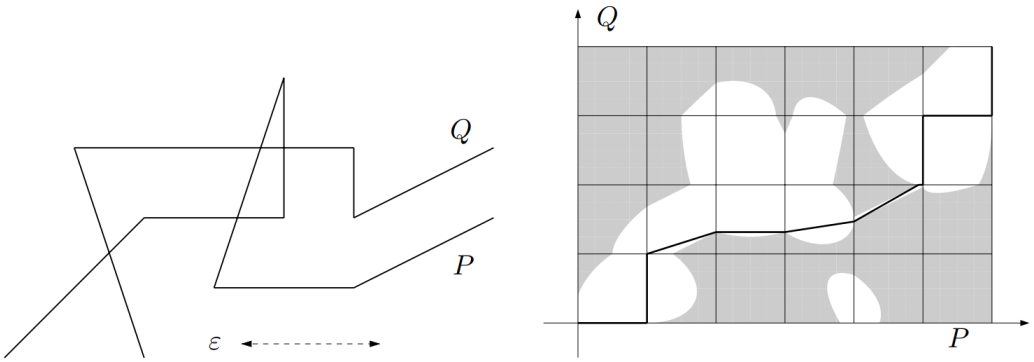


FIGURE 2.6: Free space diagram of polygonal curves P and Q (from Wenk [76]).

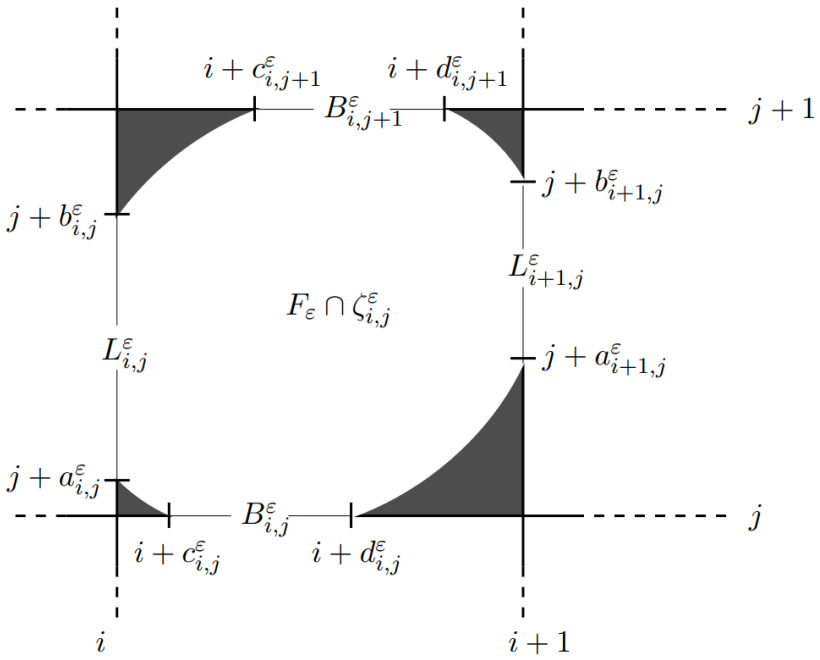


FIGURE 2.7: Cell of the free space diagram (from Wenk [76]).

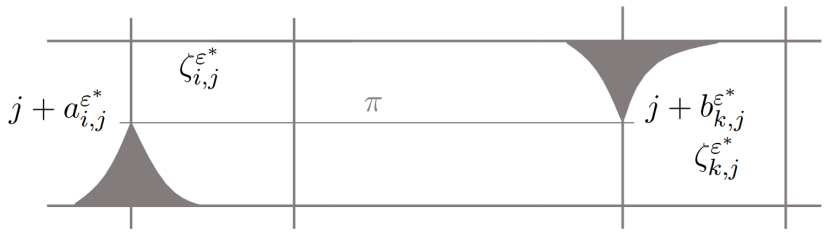
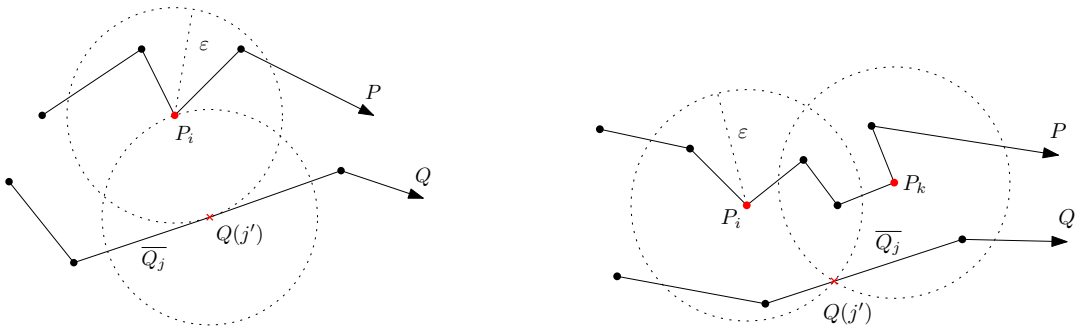


FIGURE 2.8: Clamped horizontal passage in the free space diagram (from Wenk [76]).



(A) Case 1 (B) Case 2

FIGURE 2.9: Geometric situations corresponding to horizontally clamped paths.

The authors connect the structure of the free space to the problem of computing $\delta_F(P,Q)$. They show that $\delta_F(P,Q) \leq \varepsilon$ if and only if there exists a curve within $F_\varepsilon(P,Q)$ from $(0,0)$ to (m,n) which is monotone in both coordinates. For $(i,j) \in \{0,...,m-1\} \times \{0,...,n-1\}$, let $L^\varepsilon_{i,j} = \{i\} \times [j + a^\varepsilon_{i,j}, j + b^\varepsilon_{i,j}]$ and $B^\varepsilon_{i,j} = [c^\varepsilon_{i,j}, d^\varepsilon_{i,j}] \times \{j\}$ be the left and bottom line segment respectively that bound $\zeta_{i,j} \cap F_\varepsilon$. See Figure 2.7 for an illustration. We can use a dynamic programming approach to compute those parts of the segments $L^\varepsilon_{i,j}$ and $B^\varepsilon_{i,j}$ which are reachable from $(0,0)$ by a monotone path in F_ε and thus decide if $\delta_F(P,Q) \leq \varepsilon$ by checking whether (m,n) is reachable. This can be done in $O(mn)$ time.

For the optimization problem, the authors exploit a continuity property of F_ε. If we start with $\varepsilon = 0$ and continuously increase ε, passages will open in the free space. If $\delta_F(P,Q) = \varepsilon$, then F_ε contains at least one monotone path from $(0,0)$ to (m,n). One of the following cases must then occur:

1. $L^\varepsilon_{i,j}$ or $B^\varepsilon_{i,j}$ is a single point on the path for some (i,j).
2. $a^\varepsilon_{i,j} = b^\varepsilon_{k,j}$ (or $c^\varepsilon_{i,j} = d^\varepsilon_{i,k}$) for some (i,j,k) and the path passes through $(i, j + a^\varepsilon_{i,j})$ and $(k, j + b^\varepsilon_{k,j})$ (or through $(i + c^\varepsilon_{i,j}, j)$ and $(i + d^\varepsilon_{i,k}, k)$).

In case 1, the path passes through a passage between two neighboring cells that consists of a single point. In case 2, the path contains a 'clamped' horizontal or vertical passage, illustrated in Figure 2.8. Figure 2.9 shows the geometric situations that correspond to these two cases. Note that the first case is actually a special instance of the second case where $i = k$ (or $j = k$). This observation of clamped paths leads to a finite number of critical values of ε. They consider these critical values and use the decision algorithm combined with Megiddo's parametric search technique [55] to obtain an optimization algorithm that runs in $O(mn \log(mn))$ time.

Since the introduction of this algorithm, a number of other efficient exact and approximation algorithms have been proposed and used in various applications. For the discrete Fréchet distance, Eiter and Mannila [31] introduced a simple dynamic programming algorithm to compute it in $O(mn)$ time.

Surfaces. While the definition for the (continuous) Fréchet distance given here is specifically for curves in arbitrary dimensions, the general definition is also applicable to surfaces. In contrast to curves, very little is known about computing the Fréchet distance between surfaces. Computing the Fréchet distance between triangulated surfaces is known to be NP-hard [36] and upper semi-computable [5] (i.e., there exists an algorithm that produces a sequence of real numbers that converges to the Fréchet distance), but it is not known whether it is computable. Nayyeri and Xu [58] described the first $(1+\varepsilon)$-approximation algorithm for computing the Fréchet distance between two piecewise linear surfaces of genus zero. The runtime of this algorithm is super-exponential in the size of the input and total area of the surfaces. Later, the same authors [59] show that the Fréchet distance between two piecewise linear surfaces can be decided in finite time and give the first exact algorithm for this. For the special case that one of the surfaces is a triangle, they show that the problem is in PSPACE.

Translation-invariant

The first algorithm for matching two polygonal curves in the plane with respect to the *continuous* Fréchet distance was given by Venkatasubramanian [72] for translations along a fixed direction, which runs in $O((mn)^2(m+n)\log(mn))$ time. Alt *et al.* [8] later provided an algorithm that works for translations in arbitrary directions, which uses a different approach. Similarly to the algorithm of Alt and Godau [6] for computing the Fréchet distance, they first solve the decision problem in $O((mn)^3(m+n)^2)$ time and apply Cole's trick for parametric search based on sorting to solve the optimization problem in $O((mn)^3(m+n)^2\log(m+n))$ time. This was later extended to higher dimensions by Wenk [76]. Moreover, they observe that each reference point for the Hausdorff distance with respect to a set of transformations $\mathcal{T}$ is also a reference point for the Fréchet distance with respect to $\mathcal{T}$ of the same quality. Another observation they make is that the Fréchet distance is at least the distance between the start points of the two curves. This leads to a new reference point for the Fréchet distance of curves with respect to translations of quality 1 and thus they show that substantially better reference points exist for the Fréchet distance compared to the Hausdorff distance. Using a grid-based approach to reduce the approximation factor to $1+\varepsilon$, the runtime of the algorithm finally comes down to $O(\varepsilon^{-2}mn)$.

Due to the reduced computational complexity of the *discrete* Fréchet distance, its translation-invariant version received more attention than that of the continuous Fréchet distance. The discrete Fréchet distance under translation was first studied by Jiang *et al.* [44]. They provide an exact algorithm that runs in $O(m^3n^3\log(m+n))$ time and an $(1+\varepsilon)$-approximation algorithm that runs in $O(m^3n^3\log(1/\varepsilon))$ time. This exact algorithm was later improved by Avraham *et al.* [10] and runs in $O(m^3n^2(1+\log(n/m))\log(m+n))$ time (assuming $m \leq n$). Mosig and Clausen [57] also presented an approximation algorithm under rigid motions (translation, rotation and scaling) with an approximation factor close to 2 and that runs in $O(m^2n^2)$ time. For polygonal curves of equal length n, the current fastest exact algorithm is given by Bringmann *et al.* [16] and runs in $O(n^{14/3}\log^3(n))$ time. Similarly to many of the previously listed algorithms, this algorithm relies on an arrangement-based approach.

They also show that it cannot be improved below $n^{4-o(1)}$ assuming the Strong Exponential Time Hypothesis (SETH).

2.3.4 Earth Mover's Distance

While the Fréchet distance fixes the issue encountered with the Hausdorff distance where the course of the curves are ignored, it still only focuses on capturing the maximum distance attained during the traversal of both curves. We can make significant changes to the curves while maintaining the same Fréchet distance, which likewise may not be a desirable property in certain applications. An alternative measure that stays relatively close to the Fréchet distance, but does not suffer from this issue, is Dynamic Time Warping (DTW) which instead takes the sum of distances. However, there is not much known about the translation-invariant version due to its computational intricacy [18]. A different measure that also does not have this specific problem, but has been studied more, is the Earth Mover's Distance (EMD).

The Earth Mover's Distance is widely used in fields such as image retrieval [63] and shape matching [37]. It is proportional to the minimum amount of work required to transform one distribution into the other. This transformation process can be visualized as piles of dirt and holes that need to be filled with dirt. The heavier distribution holds the piles of dirt and the lighter distribution holds the holes that need to be filled. The amount of dirt in a pile or the capacity of the hole is given by its respective weight value. The goal is then to fill all the holes such that the total work is minimized. The work is measured by the amount of moved dirt multiplied by the distance over which it is moved. If the total weights of the distributions are equal, then all the dirt has been moved to the holes, and otherwise there will be dirt leftover. The problem of optimally moving a distribution of mass was first introduced by the French mathematician Gaspard Monge in 1781 [56], which was later reformulated by the Soviet mathematician and economist Leonid Kantorovich in 1942 [47]. It became known as the Monge-Kantorovich problem and the Earth Mover's Distance is the discrete version of it. Below we provide a formal definition.

Definition

We denote a discrete distribution A as $A = \{(a_i, w_i)\}_{i=1}^{m}$ with each $a_i \in \mathbb{R}^d$ and $w_i \geq 0$. The weight of a distribution is defined as the sum of its weights, i.e., $W^A = \sum_{i=1}^{m} w_i$.

Given two distributions $A = \{(a_i, w_i)\}_{i=1}^{m}$ and $B = \{(b_i, u_i)\}_{i=1}^{n}$, a *flow* is any matrix $F = (f_{ij}) \in \mathbb{R}^{m \times n}$. Intuitively, f_{ij} is the amount of weight at a_i which is matched to weight at b_j. A flow F is called a *feasible flow* if and only if it satisfies the following constraints:

1. $f_{ij} \geq 0, i = 1, ..., m, j = 1, ..., n$
2. $\sum_{j=1}^{n} f_{ij} \leq w_i, i = 1, ..., m$
3. $\sum_{i=1}^{m} f_{ij} \leq u_j, j = 1, ..., n$
4. $\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} = \min(W^A, W^B)$

Constraint 1 enforces non-negative flow. Constraint 2 ensures that the weight in B matched to a_i does not exceed w_i. Similarly, constraint 3 ensures that the weight in A matched to b_j does not exceed u_j. Constraint 4 forces the total amount of weight matched to be equal to the lighter distribution.

Let $\mathcal{F}(A, B)$ denote the set of all feasible flows between A and B. The work done by a feasible flow $F \in \mathcal{F}(A, B)$ in matching A and B is given by:

$$\text{WORK}(F, A, B) = \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d(a_i, b_j)$$

where $d(a_i, b_j)$ is the ground distance (e.g., Euclidean distance) between a_i and b_j. The *Earth Mover's Distance* $\text{EMD}(A, B)$ between A and B is the minimum amount of work required to match the two distributions, normalized by the weight of the lighter distribution:

$$\text{EMD}(A, B) = \frac{\min_{F \in \mathcal{F}(A,B)} \text{WORK}(F, A, B)}{\min(W^A, W^B)}$$

Note that this formulation allows for a partial matching since some weight in the heavier distribution remains unmatched. It is also important to know that the EMD is only a true metric when the distributions have equal total weight and the ground distance is a metric itself; For unequal total weight distributions, they can have a zero distance even if they are not identical (thereby violating the non-negative property) and it does not obey the triangle inequality [35].

This problem is a special instance of the the minimum cost flow problem. If the distributions are unweighted point sets (or equivalently, we have unit weights $w_i = u_j = 1$ with integer flows), it simply becomes the (partial) assignment problem.

Computation

The work minimization problem is a type of linear program known as the transportation problem, for which efficient algorithms exist to solve it, such as the transportation simplex method [39]. There are also various approximation algorithms known, such as the $(1 + \varepsilon)$-approximation by Cabello *et al.* [20] that uses a combination of geometric spanners and a minimum cost flow algorithm. This runs in nearly quadratic time.

For the assignment problem, the Hungarian algorithm [51] can be used to solve it exactly in $O(n^3)$ time for sets of equal size $n = m$. For point sets in the plane, the algorithm of Vaidya [70] computes it in $O(n^2 \log^3(n))$ time for the L_1 and L_∞ metrics. Bringmann *et al.* [19] recently generalized this result to differently-sized point sets in any dimension d and proposed an algorithm that runs in $O(n^2 \log^{d+2}(n))$ time (where $n \geq m$).

Translation-invariant

Cohen and Guibas [23] introduced the *EMD under Transformation* problem, where we try to find a transformation $t \in \mathcal{T}$ of one distribution which minimizes its EMD to another:

$$\text{EMD}_{\mathcal{T}}(A, B) = \min_{t \in \mathcal{T}} \text{EMD}(A, t(B)) = \frac{\min_{t \in \mathcal{T}, F \in \mathcal{F}(A,B)} \text{WORK}(F, A, t(B))}{\min(W^A, W^B)}$$

where $t(B)$ represents the transformation t applied to distribution B. Note that our focus is on transformations that only modify the points of a distribution and leave its weights fixed (i.e., $\mathcal{F}(A, t(B)) = \mathcal{F}(A, B)$).

They presented an iterative Flow-Transformation algorithm which alternates between finding the optimum flow for a given transformation, and the optimum transformation for a given flow. This iterative procedure has been proven to converge, but not necessarily to a global optimum. In case the distributions have equal weight and the L_2^2 (squared Euclidean) distance is considered, then there is a unique optimal translation (i.e., the translation that lines up the centroids of the point sets). This allows for an efficient exact computation. In case the distributions have equal weight and the L_1 (Manhattan) distance is considered, then there is a simple solution for computing the translation for one-dimensional points that involves cumulative distribution functions.

Klein and Veltkamp [49] used reference points to approximate the problem for the translation class. They propose a 2-approximation algorithm, but this algorithm works only on weighted point sets of equal total weight. The center of mass is in this case an (optimal) reference point with respect to affine transformations and its quality is 1. For a weighted point set $A = \{(a_i, w_i)\}_{i=1}^m$, the center of mass of A is defined as:

$$C(A) = \frac{1}{W^A} \sum_{i=1}^{m} w_i a_i$$

They furthermore prove that there does not exist an EMD-reference point for weighted point sets with unequal total weights with respect to all transformation sets that include the set of translations. In light of this, they consider another distance measure called the Proportional Transportation Distance (PTD), which was introduced by Giannopoulos and Veltkamp [35]. Let $A = \{(a_i, w_i)\}_{i=1}^m$ and $B = \{(b_i, u_i)\}_{i=1}^n$ be two weighted point sets, then the Proportial Transportation Distance is defined as:

$$\mathrm{PTD}(A, B) = \frac{\min_{F \in \mathcal{F}(A,B)} \mathrm{WORK}(F, A, B)}{W^A}$$

At first sight, it looks relatively similar to the EMD, but a feasible flow F has to satisfy constraints that are different:

1. $f_{ij} \geq 0, i = 1, ..., m, j = 1, ..., n$
2. $\sum_{j=1}^{n} f_{ij} = w_i, i = 1, ..., m$
3. $\sum_{i=1}^{m} f_{ij} = \frac{u_j W^A}{W^B}, j = 1, ..., n$
4. $\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} = W^A$

In contrast to EMD, the PTD obeys the triangle inequality. It follows that the PTD is a pseudo-metric. The center of mass is then a PTD-reference point for weighted point sets of arbitrary total weight with respect to affine transformations and quality 1.

Cabello *et al.* [20] introduced, for two weighted point sets in the plane of size m and n with $m \leq n$, a $(1 + \varepsilon)$-approximation algorithm that runs in $O((n^3 m/\varepsilon^4) \log^2(n))$ time for translations where the point sets have unequal total weight. This decreases to $O((n^2/\varepsilon^4) \log^2(n))$ time for equal total weight sets. This algorithm can furthermore

be generalized to arbitrary dimensions d with a runtime of $O((n^3 m/\varepsilon^{3d-2}) \log^2(n/\varepsilon))$. In case of unweighted and equal amount of points, then they also propose a $(1 + \varepsilon)$-approximation algorithm that runs in $O((n^{3/2}/\varepsilon^{7/2}) \log^5(n))$ time. If we have unequal amount of points instead, then there are probabilistic approximations with the same guarantees which run in $O((n^3/\varepsilon^4) \log^3(n))$ time and succeed with high probability.

For unweighted points, Bringmann *et al.* [19] have also recently proposed algorithms to compute the Earth Mover's Distance under translations for points in arbitrary dimension d for the L_1 and L_∞ metrics that run in $O(m^d n^{d+2} \log^{d+2}(n))$ time, which extends on the results obtained by Epstein *et al.* [32] who already had a $O(n^6 \log^3(n))$ time algorithm for equally sized points in the plane with respect to the L_1 metric. They furthermore show that the Earth Mover's Distance under translations with the L_2 metric can not be solved exactly in any dimension $d \geq 2$ and therefore must be approximated. The reason for this is that for any equally-sized point sets A and B where B only consists of same point $(0, 0, ..., 0)$ (the origin), the EMD is simply the geometric median of A, which has no algebraic expression for $d \geq 2$ [11] (that is, no expression using only addition, multiplication, and k-th roots).

Chapter 3

Matching Under One-Dimensional Translations

In this chapter, we study the problem of matching shapes under transformations restricted to translations along one axis from an algorithmic point of view. While there has been done quite some research already on shape matching under translation in arbitrary directions, finding a one-dimensional translation that minimizes the distance between two shapes has received less attention thus far. This has applications in the problem we are trying to solve for estimating the TDOA between sound signals for example, but also potentially in other applications where alignment of time series data is required. We start with a formal description of the problem and after that describe (multiple variations of) exact or approximation methods that solve it for three different geometric distance measures.

3.1 Problem Statement

Let $A, B \subseteq \mathbb{R}^d$ be shapes in d-dimensional space with $d \geq 1$. Without loss of generality, we only allow translations along the first component (which we sometimes refer to as the x-axis). We first introduce some notation. We denote a point as $a = (x, a_r) \in A$ where x is the first component and $a_r \in \mathbb{R}^{d-1}$ denotes the remaining components. For a point $a = (x, a_r) \in A$ and a one-dimensional translation $t \in \mathbb{R}$, we write $a + t = (x + t, a_r)$. Note that the other component(s) a_r remain unaltered. We also use

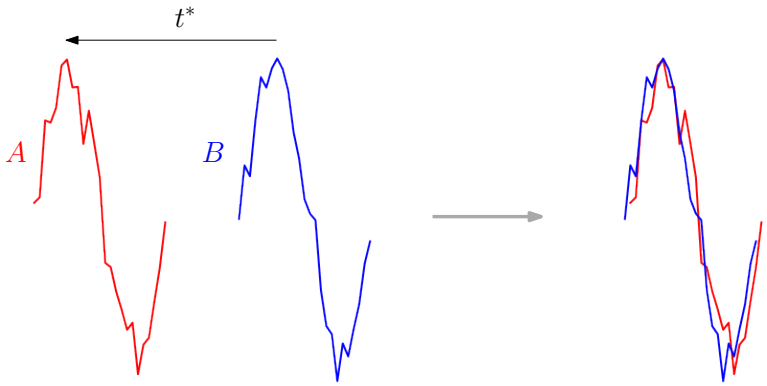


FIGURE 3.1: Matching noisy sine waves under translations along the x-axis.

this notation for a set of points: $A + t = \{a + t \mid a \in A\}$. Analogously for the set B. We adopt this notation throughout the rest of this thesis.

Let $\mathcal{T} \subseteq \mathbb{R}$ be the collection of all one-dimensional translations and $D(A, B)$ be the distance between A and B where $D : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is some distance measure. Without loss of generality, we fix the set A and only allow the set B to translate. Then we try to find a one-dimensional translation $t^* \in \mathcal{T}$ such that:

$$D(A, B + t^*) = \min_{t \in \mathcal{T}} D(A, B + t)$$

An illustration of the problem is given in Figure 3.1. It is important to note that the specific distance measure D used could significantly affect the optimal one-dimensional translation t^*.

In this formulation, the entire shapes of A and B must be optimally matched. However, we might encounter situations where we want to match A to only a part of B. This variant is known as the *partial* matching problem. In that case, we have a collection of partial shapes $\mathcal{B} \subseteq \mathcal{P}(B)$ (where $\mathcal{P}(B)$ denotes the power set of B), and we then try to find a partial shape $B^* \in \mathcal{B}$ and a one-dimensional translation $t^* \in \mathcal{T}$ such that:

$$D(A, B^* + t^*) = \min_{B' \in \mathcal{B}, t \in \mathcal{T}} D(A, B' + t)$$

In this thesis, we will mainly focus on developing algorithms that solve the regular matching problem. Note that we can still solve the partial matching problem by computing a regular matching for each $B' \in \mathcal{B}$ and taking the minimum distance of those. In that case, the runtime would be proportional to the size of $\mathcal{B}$ (which may be infinite).

3.2 Hausdorff Distance

For two compact sets A and B, we define $H^*(A, B)$ as the minimum Hausdorff distance between A and B over all one-dimensional translations $t \in \mathbb{R}$ of B:

$$H^*(A, B) = \min_{t \in \mathbb{R}} H(A, B + t)$$

Inspired by the approach of Huttenlocher *et al.* [41], we will also analyze the structure of the upper envelope for this specific problem. Let $A = \{a_1, ..., a_m\}$ and $B = \{b_1, ..., b_n\}$ be two compact sets in $\mathbb{R}^d$. We denote the distance between a pair of entities $a_i \in A$ and $b_j \in B$ (which may be points, line segments or triangles for example), as $b_j \in B$ undergoes a one-dimensional translation t, by:

$$\delta_{ij}(t) = \rho(a_i, b_j + t)$$

where $\rho(a_i, b_j)$ is the (static) distance between the two entities a_i and b_j. We define the function $d_i(t)$ as the lower envelope of the functions $\delta_{ij}(t)$ for a given $a_i \in A$ and all $b_j \in B$:

$$d_i(t) = \min_{b_j \in B} \delta_{ij}(t)$$

Similarly, $d'_j(t)$ is the lower envelope for a given $b_j \in B$ and all $a_i \in A$. We define the function $f(t)$ as the upper envelope of the functions $d_i(t)$ for each $a_i \in A$:

$$f(t) = \max_{a_i \in A} d_i(t)$$

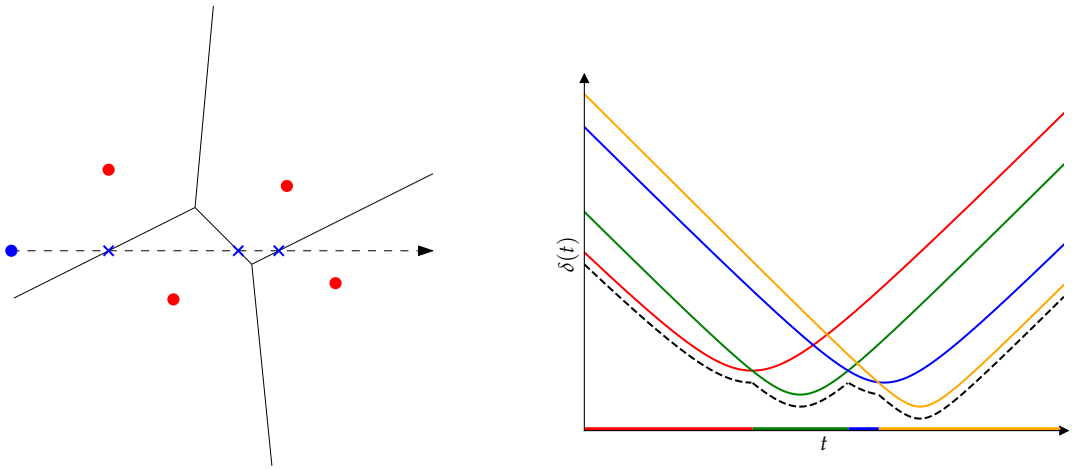


FIGURE 3.2: Example of the lower envelope $d'_j(t)$ for point sets (with ρ being the L_2 metric).

Similarly, $f'(t)$ is the upper envelope of the functions $d'_j(t)$ for each $b_j \in B$. Note that $f(t)$ and $f'(t)$ specify the values of $h(A, B+t)$ and $h(B+t, A)$ as a function of the translation t, respectively. Therefore:

$$H^*(A,B) = \min_{t\in\mathbb{R}} \max\{f(t), f'(t)\}$$

An illustration of a lower envelope $d'_j(t)$ for some $b_j \in B$ is given in Figure 3.2. The envelopes are formed by a set of *curved edges* where each edge belongs to the graph of a single function $\delta_{ij}(t)$. The endpoints of the edges are located at the intersection points of these functions and are called the *vertices* of the envelope. Note that $\delta_{ij}(t)$ itself might also have vertices depending on its definition (e.g., the point-to-segment function in the plane is composed of edges from the continuous point-to-point and point-to-line functions).

3.2.1 Algorithm

We describe the algorithm for computing $H^*(A,B)$ and its corresponding translation below step-by-step. It uses the algorithm that is provided by Boissonnat and Yvinec [14] for computing the lower envelope of univariate functions. Note that we do not have to explicitly construct $f(t)$ and $f'(t)$ first to compute $H^*(A,B)$ since we can directly compute the upper envelope of all the lower envelopes.

1. Compute the lower envelope $d_i(t)$ for a given $a_i \in A$. This can be done using a divide-and-conquer approach [14].
   - Recursively split the set of n functions $\delta_{ij}(t)$ into two halves until we reach a single function. By definition, the lower envelope of a single function is the function itself.
   - In each merge step, we compute the combined lower envelope from the two (partial) lower envelopes $\mathcal{I}_1$ and $\mathcal{I}_2$. Due to the x-monotonicity property of these envelopes, we can achieve this by a plane sweep from left to right where at any time the sweep line intersects an edge of $\mathcal{I}_1$ and $\mathcal{I}_2$. When it passes over a vertex or an intersection point, the edge that contributed to the combined lower envelope simply becomes a new edge in this envelope.

Do this for all m functions $d_i(t)$.

2. Apply the same procedure to all n functions $d'_j(t)$.
3. Compute the upper envelope of the $m + n$ lower envelopes from the previous two steps. This can simply be done using the same divide-and-conquer approach as previously, but the goal is then to find an upper envelope instead.
4. The global minimum then either lies on a vertex of the envelope or at a local minimum of an edge.

The time complexity of this algorithm depends on the combinatorial complexity of the envelopes, which can be determined using Davenport-Schinzel sequences [1]. It follows that the lower envelope of a set of n functions which intersect each other at most s times has complexity $O(\lambda_s(n))$ and can be computed in $O(\lambda_s(n)\log(n))$ time [14], where it has been shown that

$$\begin{aligned}\lambda_1(n) &= n\\ \lambda_2(n) &= 2n-1\\ \lambda_3(n) &= \Theta(n\alpha(n))\end{aligned}$$

with $\alpha(n)$ being the inverse Ackermann function. For $s > 3$, it is still nearly linear in n.

Suppose that each pair of functions $\delta_{ij}(t)$ intersect each other at most s times. The first step then takes $O(m \cdot \lambda_s(n)\log(n))$ time and the second step, analogously, takes $O(n \cdot \lambda_s(m)\log(m))$ time. The third step has a recursion depth of $O(\log(m+n))$ and the complexity of this upper envelope is $O(\lambda_s(mn))$, so the runtime comes down to $O(\lambda_s(mn)\log(m+n))$. The fourth step is a simple iteration over the vertices and local minima, which takes $O(\lambda_s(mn))$ time. This brings us to the following theorem:

Theorem 1. *Given two compact sets A and B of sizes m and n and a pairwise distance function $\delta_{ij}(t)$ for each $a_i \in A$ and $b_j \in B$ whose graphs intersect each other at most s times, then $H^*(A,B)$ and its corresponding one-dimensional translation t can be computed in $O(\lambda_s(mn)\log(m+n))$ time.*

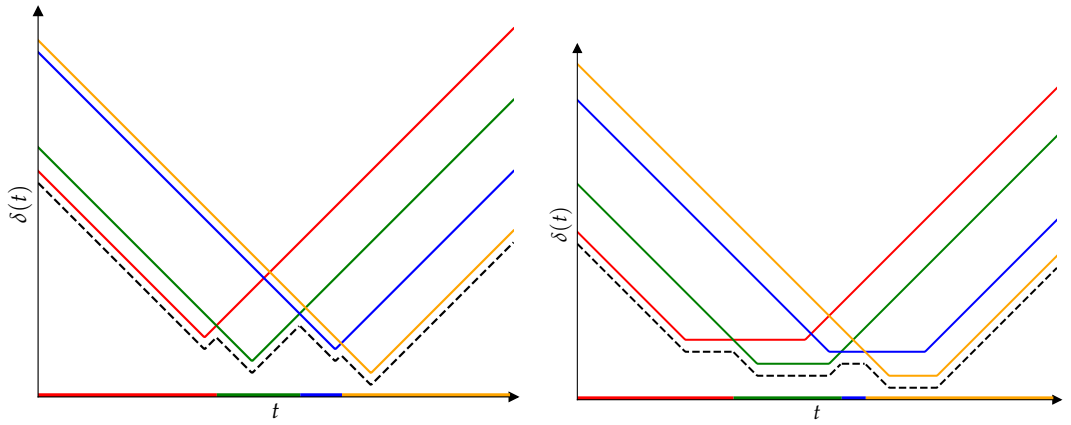
Note that the runtime simply depends on how the distance function $\delta_{ij}(t)$ between each entity $a_i \in A$ and $b_j \in B$ is defined, which determines the bound on the number of intersections s between them. In the next sections, we will analyze the case where A and B are point sets in any dimension. We then extend the analysis to line segments in the plane and triangles in three-dimensional space.

3.2.2 Point sets

For a point $a_i \in A$, we denote by a_i^k the k-th component of a_i (similarly for a point $b_j \in B$). Note that for any L_p-norm with $p \geq 1$, the function $\delta_{ij}(t)$ becomes of the following form:

$$\begin{aligned}\delta_{ij}(t) &= (|a_i^1 - (b_j^1 + t)|^p + \sum_{k=2}^{d} |a_i^k - b_j^k|^p)^{1/p}\\ &= (|x-t|^p + c)^{1/p}\end{aligned}$$

where $x = a_i^1 - b_j^1$ and $c = \sum_{k=2}^{d} |a_i^k - b_j^k|^p$ are constants. We prove the following property:



(A) Manhattan (L_1) (B) Chebyshev (L_∞)

FIGURE 3.3: Example of the lower envelope $d'_j(t)$ for point sets with different metrics as ρ (and for the same set of points as in Figure 3.2).

Proposition 1. *Two functions $\delta_{ij}(t)$ intersect each other at most once.*

Proof. Let $f(t) = (|x_1 - t|^p + c_1)^{1/p}$ and $g(t) = (|x_2 - t|^p + c_2)^{1/p}$ be two such functions. The intersection is given by:

$$\begin{aligned}(|x_1 - t|^p + c_1)^{1/p} &= (|x_2 - t|^p + c_2)^{1/p} \\ |x_1 - t|^p + c_1 &= |x_2 - t|^p + c_2 \\ |x_1 - t|^p - |x_2 - t|^p &= c_2 - c_1\end{aligned}$$

Let us analyze the function $h(t) = |x_1 - t|^p - |x_2 - t|^p$. We can assume that, in general, $x_1 \neq x_2$. Suppose that $x_1 < x_2$. The points that change one of the functions is at $t = x_1$ and $t = x_2$.

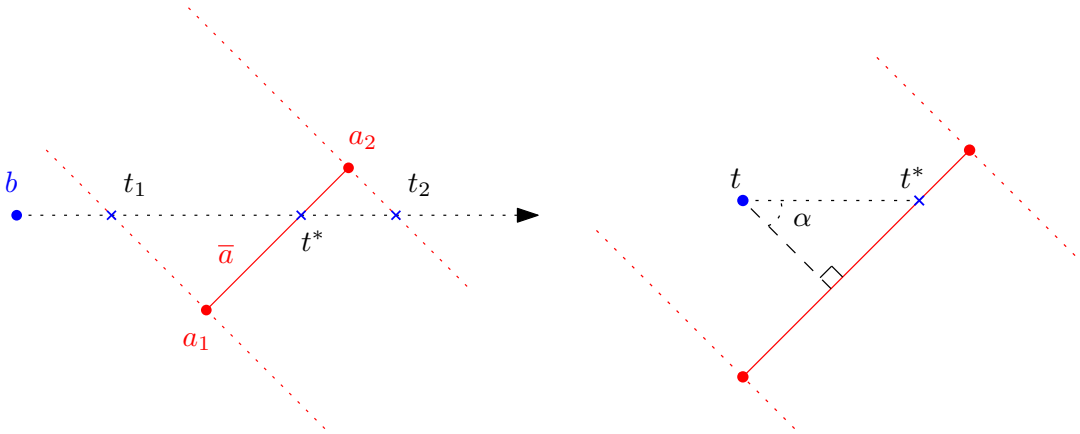
Consider the interval $t < x_1$. This means that $|x_1 - t|^p = (x_1 - t)^p$ and $|x_2 - t|^p = (x_2 - t)^p$. As t increases, they both decrease (and $(x_1 - t)^p$ decreases faster for $p > 1$). So $h(t)$ is non-decreasing (and strictly increasing for $p > 1$).

Consider now the interval $x_1 \leq t \leq x_2$. We get $|x_1 - t|^p = (-x_1 + t)^p$, so this is increasing as t grows, whereas $(x_2 - t)^p$ is still decreasing. This means that $h(t)$ is (strictly) increasing.

For the interval $t > x_2$, we get $|x_2 - t|^p = (-x_2 + t)^p$, which also starts increasing as t grows (again, $(-x_1 + t)^p$ increases faster for $p > 1$). So $h(t)$ is again non-decreasing (and strictly increasing for $p > 1$).

Therefore, $h(t)$ is a monotonically increasing function (for $p > 1$, it is even strictly monotone). For $p = 1$, there is at most one intersection at $h(t) = c_2 - c_1$ and, for $p > 1$, we have exactly one intersection. A similar argument holds for $x_1 > x_2$ where $h(t)$ is a monotonically decreasing function instead. □

An illustration of the lower envelope $d_i(t)$ for some $a_i \in A$ is given in Figure 3.2 and 3.3 for the L_1, L_2 and L_∞ metrics, which are the metrics that we are especially interested in. It is easy to see that the number of vertices is at most $n - 1$: a single function has no vertices and each additional function adds at most one vertex at an intersection point. This is in accordance with the Davenport-Schinzel sequence where the number of vertices is at most $\lambda_1(n) = n$. The complexity of the upper envelope is thus $O(\lambda_1(mn)) = O(mn)$. By Proposition 1 and Theorem 1, we immediately obtain the following lemma:



(A) Event points (B) Perpendicular distance

FIGURE 3.4: Point-to-segment distance under translation.

Lemma 1. *Given two point sets A and B of sizes m and n in any (constant) dimension, then $H^*(A, B)$ and its corresponding one-dimensional translation t can be computed in $O(mn \log(m+n))$ time for any L_p-norm where $p \geq 1$.*

3.2.3 Line segments in plane

We now consider the case where A and B are sets of line segments in $\mathbb{R}^2$. To use the algorithm that we described previously, we need to define a distance function

$$\delta_{ij}(t) = \text{seg-to-seg}(\overline{a}_i, \overline{b}_j + t)$$

for each segment $\overline{a}_i \in A$ and $\overline{b}_j \in B$ where seg-to-seg$(\overline{a}_i, \overline{b}_j + t)$ denotes the distance between them when $\overline{b}_j$ is translated by t. In the sequel, when we refer to seg-to-seg (or any other distance between two entities) as a function, we mean the function over the translation t. Also, we will be using the Euclidean distance as underlying metric.

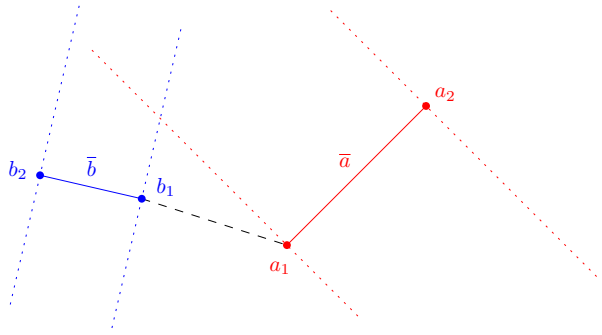
It is easy to see that for two non-intersecting line segments, the shortest distance is always defined on one of the endpoints. With that in mind, let us first focus on the distance between a point b and a line segment $\overline{a}$ where b is moved along the x-axis.

Proposition 2. *The point-to-segment distance function is given by:*

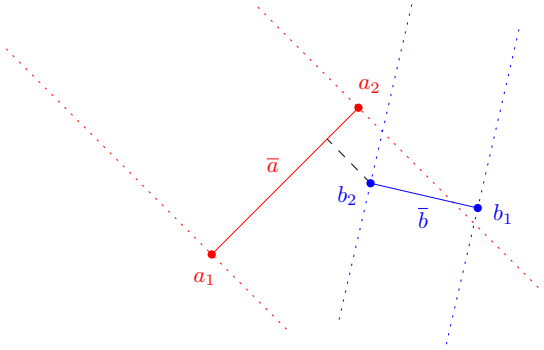
$$\textit{point-to-seg}(b+t, \overline{a}) = \begin{cases} d(a_1, b+t) & \text{if } t \leq t_1 \\ \cos(\alpha) \cdot |t - t^*| & \text{if } t_1 < t \leq t_2 \\ d(a_2, b+t) & \text{if } t > t_2 \end{cases}$$

where d denotes the Euclidean distance, of which the function over the translation t has already been covered with point sets, and α is the angle between the horizontal line and perpendicular line (see Figure 3.4b).

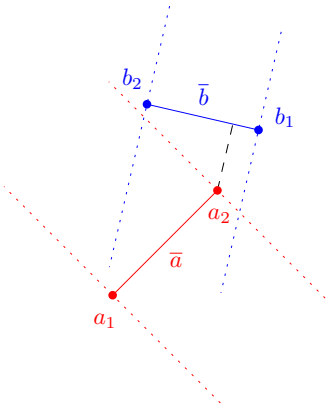
Proof. If we decompose $\overline{a}$ into its edge and two vertices a_1 and a_2, we can create a Voronoi diagram which divides the plane into three areas: the points closest to the endpoints and the points closest to the interior of the line segment (see Figure 3.4a). We call the area of the points closest to the interior the interior cell of $\overline{a}$. Observe that outside the interior cell, the distance between the point and the line segment is simply the Euclidean distance between the point and the closest endpoint. Within the interior cell, the distance is defined by the perpendicular distance (see Figure



(A) Outside interior cell



(B) Case 1



(C) Case 2

FIGURE 3.5: Segment-to-segment distance under translation.

3.4b). We call the translations where a point b leaves and enters another area the event points. In the general case, these are the translations t_1 and t_2 where b enters and leaves the interior cell respectively. There exists exactly two such translations due to the convexity property of the areas, so it may enter and leave an area at most once. We also have the translation t^* where b intersects $\overline{a}$ (or its infinite line). The distance function point-to-seg$(b + t, \overline{a})$ is then a piecewise function which changes at the event points. □

Note that we also have the degenerate cases with the line segment either being horizontal, where the distance simply becomes constant when it enters the interior cell until it leaves the interior cell again, or vertical where the point either stays within or outside the interior cell. Also note that if the line segment is the moving entity instead, the event points are simply the negation.

The distance between two line segments $\overline{a}$ and $\overline{b}$ is then a piecewise function composed of the point-to-segment distances:

Proposition 3. *The segment-to-segment distance function is given by:*

$$\textit{seg-to-seg}(\overline{a}, \overline{b} + t) = \begin{cases} 0 & \textit{if they intersect} \\ \min\{\textit{point-to-seg}(a_1, \overline{b} + t), & \\ \quad \textit{point-to-seg}(a_2, \overline{b} + t), & \\ \quad \textit{point-to-seg}(b_1 + t, \overline{a}), & \\ \quad \textit{point-to-seg}(b_2 + t, \overline{a})\} & \textit{otherwise} \end{cases}$$

and, by analysis of the behavior of this function, consists of at most four vertices.

Proof. For simplicity, we ignore the degenerate cases and assume for now that the line segments are not parallel, perpendicular, horizontal or vertical. Similarly with $\overline{a}$, we create a Voronoi diagram of the edge and vertices of $\overline{b}$. We first observe that when both line segments are outside the interior cells, then by definition the distance between them is at the endpoints. Specifically, the endpoint that defines the area where the other line segment resides in. Without loss of generality, let this be the endpoints a_1 and b_1. We observe that this changes once the line segments enter the interior cells of each other. How it specifically changes depends on the orientation and position relative to each other, but we can assume that w.l.o.g. again b_1 is the first endpoint that enters the interior cell of $\overline{a}$. This means that the distance is first defined by point-to-seg$(b_1 + t, \overline{a})$ (Figure 3.5a) until either

1. b_1 intersects with $\overline{a}$, where the distance becomes zero, or
2. when a_2 enters the interior cell of $\overline{b}$.

If (1) occurs, then the distance remains zero until either b_2 intersects $\overline{a}$ where the distance becomes point-to-seg$(b_2 + t, \overline{a})$ (Figure 3.5b), or a_2 intersects $\overline{b}$. In the latter case or when (2) occurs, the distance becomes point-to-seg$(a_2, \overline{b} + t)$ (Figure 3.5c). There are no more switches in the distance function after that. This means that the function is composed of at most three (Euclidean) endpoint-to-endpoint distances and two (perpendicular) endpoint-to-segment distances. It is easy to see that this also holds for the degenerate cases. □

We now have a well-defined distance function $\delta_{ij}(t)$ for two line segments $\overline{a}_i \in A$ and $\overline{b}_j \in B$. Moreover, it can be observed that each pair of the individual continuous endpoint-to-endpoint and endpoint-to-segment distance functions intersect each other at most twice. The complexity of the upper envelope is therefore $O(\lambda_2(5mn)) = O(mn)$. Given this observation and Theorem 1, we obtain the following lemma:

Lemma 2. *Given two sets of line segments A and B of sizes m and n in the plane, then $H^*(A, B)$ and its corresponding one-dimensional translation t can be computed in $O(mn \log(m + n))$ time.*

3.2.4 Triangles in 3D space

Finally, we consider the case where A and B are triangles in $\mathbb{R}^3$. Similarly to line segments, we need to define a distance function

$$\delta_{ij}(t) = \triangle\text{-to-}\triangle(\hat{a}_i, \hat{b}_j + t)$$

for each triangle $\hat{a}_i \in A$ and $\hat{b}_j \in B$ where $\triangle$-to-$\triangle(\hat{a}_i, \hat{b}_j + t)$ denotes the distance between them when $\hat{b}_j$ is translated by t. We again take the Euclidean distance as underlying metric.

It is easy to see that the shortest distance between two non-intersecting triangles in three-dimensional space either involves one of vertices or a pair of edges between the triangles. There are six vertex-triangle pairs and nine edge-edge pairs. Let us start with the vertex-triangle distance and after that the edge-edge distance.

Vertex-Triangle Distance. If we take the Voronoi diagram of the vertices and edges of the triangle, we divide the space into exactly seven areas: the points closest to the interior, the points closest to each of the edges and the points closest to each of the

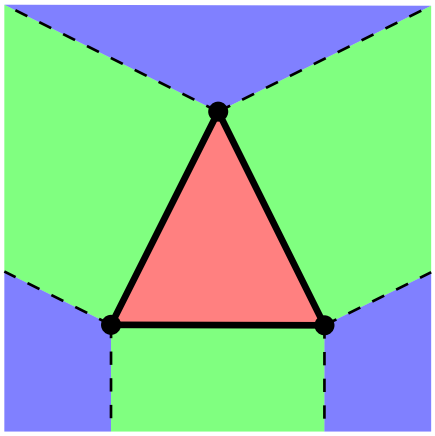


FIGURE 3.6: Subdivision of areas for a triangle.

vertices. See Figure 3.6 for an illustration of the two-dimensional projection where each area category has been assigned a different color. The blue areas are simply the point-to-point Euclidean distances, which we have already seen with point sets. The green areas are the point-to-line distances where the line is defined by the respective line segment.

Proposition 4. *Given a point p and a line defined by the line segment $\overline{ab}$ with endpoints a and b, the point-to-line distance function is of the form (with constants c):*

$$\textit{point-to-line}(p + t, \overline{ab}) = c_1 \cdot \sqrt{c_2 + (c_3 + c_4 t)^2 + (c_5 + c_6 t)^2}$$

Proof. Let $p = (x_0, y_0, z_0)$, $a = (x_1, y_1, z_1)$, and $b = (x_2, y_2, z_2)$. The static point-to-line distance is given by:

$$\text{point-to-line}(p, \overline{ab}) = \frac{||ap \times ab||}{||ab||} = \frac{\sqrt{c_x^2 + c_y^2 + c_z^2}}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}}$$

where $c_x = (y_0 - y_1)(z_2 - z_1) - (z_0 - z_1)(y_2 - y_1)$ and analogously for c_y and c_z, which are simply the cross product components. Note the use of vector notation where for example $ab = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$ (the vector from a to b) and $||ab||$ is the magnitude of the vector. The interpretation of this formula is as follows: The cross-product of ap and ab produces a vector that is perpendicular to both vectors and its magnitude is the area of the parallelogram formed by ap and ab. The height of the parallelogram is the perpendicular distance from p to ab, which we obtain by dividing the area with the base, which is simply the magnitude of ab. Defining this as a function over the translation along the x-axis for point p proves the proposition. □

We then also have the red area that is the point-to-plane distance where the plane is defined by the interior of the triangle.

Proposition 5. *Given a point p and the plane defined by the triangle $\triangle abc$ with vertices a, b and c, the point-to-plane distance function is of the form (with constants c):*

$$\textit{point-to-plane}(p + t, \triangle abc) = c_1 \cdot |c_2 t + c_3|$$

Proof. Let $p = (x_0, y_0, z_0)$, $a = (x_1, y_1, z_1)$, $b = (x_2, y_2, z_2)$, and $c = (x_3, y_3, z_3)$. The normal vector of the plane (which is simply the perpendicular vector to the plane) is given by:

$$n = ab \times ac = (n_x, n_y, n_z)$$

where $n_x = (y_2 - y_1)(z_3 - z_1) - (z_2 - z_1)(y_3 - y_1)$ and analogously for n_y and n_z. The static point-to-plane distance is given by:

$$\text{point-to-plane}(p, \triangle abc) = \frac{|ap \cdot n|}{||n||} = \frac{|n_x(x_0 - x_1) + n_y(y_0 - y_1) + n_z(z_0 - z_1)|}{\sqrt{n_x^2 + n_y^2 + n_z^2}}$$

The interpretation is again as follows: Note that the shortest distance from a point to plane is along the direction of the normal vector. This means that we need to project the vector ap onto n. The dot product of ap and n measures how much ap points along n and dividing it by the magnitude of the normal vector normalizes the projection. Since the dot product can be either positive (point is above the plane, as defined by the normal vector) or negative (point is below the plane), we also need to take the absolute value. Defining this as a function over the translation for point p proves the proposition. □

This means that the point-to-triangle distance function point-to-$\triangle(p + t, \triangle abc)$ for a point p and triangle $\triangle abc$ is determined by the area where p resides in when it is translated by t. It follows that:

Proposition 6. *The point-to-triangle distance function consists of at most six vertices.*

Proof. Due to the convexity of the Voronoi cells, a point that is translated along the x-axis will enter and leave the same cell at most once. This means that the number of switches in the distance function is bounded by the number of areas, of which there are seven in total (see Figure 3.6). □

Edge-Edge Distance. This setting is relatively similar to line segments in the plane where the Voronoi diagram of the vertices and edge of a line segment in three-dimensional space results in a division into three areas. However, in contrast to that setting, the line segments are generally not coplanar, which makes the analysis of how the distance function behaves a bit more difficult. We again have the point-to-point and point-to-line distances, which we already discussed for the vertex-triangle distance, but we now additionally have a line-to-line distance.

Proposition 7. *Given two lines defined by the line segment $\overline{ab}$ with endpoints a and b, and the line segment $\overline{cd}$ with endpoints c and d, the line-to-line distance function is of the form (with constants c):*

$$\textit{line-to-line}(\overline{ab}, \overline{cd} + t) = c_1 \cdot |c_2 t + c_3|$$

Proof. Let $a = (x_1, y_1, z_1)$, $b = (x_2, y_2, z_2)$, $c = (x_3, y_3, z_3)$, and $d = (x_4, y_4, z_4)$. The vector perpendicular to both lines is given by:

$$n = ab \times cd = (c_x, c_y, c_z)$$

where $c_x = (y_2 - y_1)(z_4 - z_3) - (z_2 - z_1)(y_4 - y_3)$ and analogously for c_y and c_z. The static line-to-line distance is then given by:

$$\text{line-to-line}(\overline{ab}, \overline{cd}) = \frac{|ac \cdot n|}{||n||} = \frac{|c_x(x_3 - x_1) + c_y(y_3 - y_1) + c_z(z_3 - z_1)|}{\sqrt{c_x^2 + c_y^2 + c_z^2}}$$

The idea is relatively similar to the point-to-plane distance. The shortest distance between the two lines is a line segment that is perpendicular to both lines, so we need to project any vector between the two lines, in our case ac, onto n. Defining this as a function over the translation for $\overline{cd}$ proves the proposition (which as expected, is of the same form as the point-to-plane distance function). □

Note that the line-to-line distance occurs when the shortest distance between $\overline{ab}$ and $\overline{cd}$ is defined within their interiors. This is the case when they are perpendicular to each other. Recall that the vector n is the vector that is perpendicular to both lines. We can then obtain the normal vector of the plane that goes through the segment $\overline{ab}$ and is perpendicular to $\overline{cd}$ by:

$$n_{ab} = n \times ab$$

And analogously we obtain n_{cd}. We observe that when we translate $\overline{cd}$, one of the endpoints will first intersect the plane defined by n_{ab}. Up to the point where the other endpoint intersects the plane, the segment remains perpendicular to the line defined by $\overline{ab}$. If at the same time one of the endpoints of $\overline{ab}$ also first intersects the plane of n_{cd}, then $\overline{ab}$ is also perpendicular to the line defined by $\overline{cd}$. When they both meet this condition, then the shortest distance is defined by the line-to-line distance.

By analyzing the behavior of the (now three-dimensional) segment-to-segment distance function seg-to-seg-3D$(\overline{ab}, \overline{cd} + t)$, we find the following:

Proposition 8. *The segment-to-segment distance function consists of at most 12 vertices.*

Proof. The previous observation suggests that we need to keep track of a state that defines the position of the segments relative to each other and such that given a state, we can determine which function defines the shortest distance. The state changes at certain translations, which we call the event points. For each endpoint of one segment, we store in which of the three areas (endpoint 1, interior or endpoint 2) it resides of the other segment. Each endpoint has two translations for entering and leaving the interior cell, which creates a total of eight event points. For each segment, we also store which endpoint is closest to the plane that is perpendicular to the segment and goes through the other segment, or whether the segment is currently intersecting this plane. This creates another four event points. So we have a total of 12 event points that alter the state and potentially change the distance function. □

Using this defined state, if both segments are currently in the intersecting state, then the distance is defined by their lines. If two endpoints are not in an interior cell and are each other's closest endpoints, then the distance is defined between these two points. If an endpoint is within the interior cell of the other segment and it is also closest to the perpendicular plane through this segment, then the distance is defined between this point and line.

Triangle-Triangle Distance. Let $vert(\hat{a})$ and $edge(\hat{a})$ denote the set of vertices and edges respectively of a triangle $\hat{a}$. With some slight abuse of notation, for two triangles $\hat{a}$ and $\hat{b}$, let point-to-$\triangle(vert(\hat{a}), \hat{b})$ denote all vertex-triangle distances for each vertex in $\hat{a}$, and let seg-to-seg-3D$(edge(\hat{a}), edge(\hat{b}))$ denote all edge-edge distances between each pair of edges of $edge(\hat{a})$ and $edge(\hat{b})$. We can now formally specify the following:

Proposition 9. *The triangle-to-triangle distance function is given by:*

$$\triangle\text{-to-}\triangle(\hat{a}, \hat{b}+t) = \begin{cases} 0 & \text{if they intersect} \\ \min\{\text{point-to-}\triangle(vert(\hat{a}), \hat{b}+t), & \\ \quad \text{point-to-}\triangle(vert(\hat{b}+t), \hat{a}), & \\ \quad \text{seg-to-seg-3D}(edge(\hat{a}), edge(\hat{b}+t))\} & \text{otherwise} \end{cases}$$

and consists of at most 85 *vertices.*

Proof. One can easily observe that each pair of the individual continuous distance functions (point-to-point, point-to-line, point-to-plane and line-to-line) intersect each other at most twice again. Since there are nine point-to-point, 18 point-to-line, six point-to-plane and nine line-to-line distance functions (a total of 42), a naive bound on the number of vertices in the lower envelope would therefore be $\lambda_2(42) = 83$. Moreover, it is clear that two triangles only have at most two intersection events for entering and leaving the intersection region due to their convex shapes, which creates at most two additional vertices. □

So we now also obtain a well-defined distance function $\delta_{ij}(t)$ for two triangles $\hat{a}_i \in A$ and $\hat{b}_j \in B$. The complexity of the upper envelope is again $O(\lambda_2(42mn)) = O(mn)$ and the lemma follows by Theorem 1:

Lemma 3. *Given two sets of triangles A and B of sizes m and n in $\mathbb{R}^3$, then $H^*(A, B)$ and its corresponding one-dimensional translation t can be computed in $O(mn \log(m+n))$ time.*

It is worth noting that the time complexity for line segments and triangles is still very similar compared to point sets in $\mathbb{R}^d$ for any $d \geq 1$, where it was already shown that their dimension for $d > 1$ is irrelevant to the complexity. The results we obtained for line segments and triangles also suggest that, even for more complex geometric entities in arbitrary dimensions, the complexity largely depends on the degree of freedom of the translation, which is a fixed parameter in our problem.

3.2.5 2-approximation

For the minimal x and y coordinates of a set A, denoted by x_{min} and y_{min} respectively, Alt *et al.* [4] showed that the point (x_{min}, y_{min}) is a reference point of quality $\sqrt{2}$ for translations in arbitrary directions, which results in a $(1+\sqrt{2})$-approximation. Although the concept of a reference point does not exactly translate well to this specific problem, we take inspiration from this approach and show a 2-approximation for translations along the x-axis.

For two compact sets A and B, let x_A and x_B be their lowest x-coordinates. Let t_x be the translation that matches x_A and x_B, that is, $t_x = x_A - x_B$. We have the following theorem:

Theorem 2. *$H(A, B+t_x) \leq 2 \cdot H^*(A, B)$ if the underlying metric is any L_p-norm with $p \geq 1$.*

Proof. Let t^* denote an optimal translation and let x_{B+t^*} denote the lowest x-coordinate of $B + t^*$. By definition, it is clear that

$$|x_A - x_{B+t^*}| \leq H(A, B+t^*)$$

Since $x_{B+t_x} = x_A$, we can obtain $B + t^*$ from $B + t_x$ by the translation $x_{B+t^*} - x_A$. It is easy to see that any point in $B + t^*$ has a point $B + t_x$ with distance $|t^* - t_x|$ (and vice versa). Therefore, we have

$$\begin{aligned} H(B + t^*, B + t_x) &\leq |x_{B+t^*} - x_A| \\ &\leq H(A, B + t^*) \end{aligned}$$

By the triangle inequality, we have

$$\begin{aligned} H(A, B + t_x) &\leq H(A, B + t^*) + H(B + t^*, B + t_x) \\ &\leq 2 \cdot H(A, B + t^*) \end{aligned}$$

□

Note that this also holds if we take the highest x-coordinates instead.

3.3 Fréchet Distance

In this section, we consider both the continuous and discrete Fréchet distance between polygonal curves in the plane. We will not be considering the Fréchet distance for surfaces due to its computational intricacy.

3.3.1 Continuous

For polygonal curves P and Q consisting of m and n segments respectively, we define $\delta_F^*(P, Q)$ as the minimum continuous Fréchet distance between P and Q over all one-dimensional translations $t \in \mathbb{R}$ of Q:

$$\delta_F^*(P, Q) = \min_{t \in \mathbb{R}} \delta_F(P, Q + t)$$

Alt *et al.* [8] provided an exact algorithm for matching polygonal curves in the plane under translations with respect to the continuous Fréchet distance. Wenk [76] later generalized this approach to any dimension and transformations of arbitrary degree of freedom. We will describe this method specifically for translations restricted to one dimension by adapting [8].

Let $t \in \mathbb{R}$ be a one-dimensional translation. The free space of two line segments is the intersection of the unit square with an ellipse (see Figure 2.6). Consider all mn ellipses, each corresponding to a pair of line segments from P and Q. It is easy to see that each ellipse in $F_\varepsilon(P, Q + t)$ is a translation of its corresponding ellipse in $F_\varepsilon(P, Q)$. Therefore, each ellipse varies continuously in $t \in \mathbb{R}$. This also holds for ellipses that have an empty intersection with the unit square and for degenerate ellipses (i.e., the space between parallel line segments). We already observed earlier that each ellipse in F_ε varies continuously in ε. The lemma follows:

Lemma 4. *For two given polygonal curves P, Q, a one-dimensional translation $t \in \mathbb{R}$ and $\varepsilon > 0$, we have that $F_\varepsilon(P, Q + t)$ varies continuously in ε and t.*

A general proof for this is given by Wenk [76]. This continuity property plays an important role in the algorithm described below. First we introduce the definitions of configurations and critical translation sets, which are also similarly described in [8].

Definition 1 (Configurations). *A triple $(P_i, P_k, \overline{Q_j})$ for vertices P_i, P_k of polygonal curve P and line segment $\overline{Q_j}$ of polygonal curve Q is called an h-configuration. Analogously, a triple*

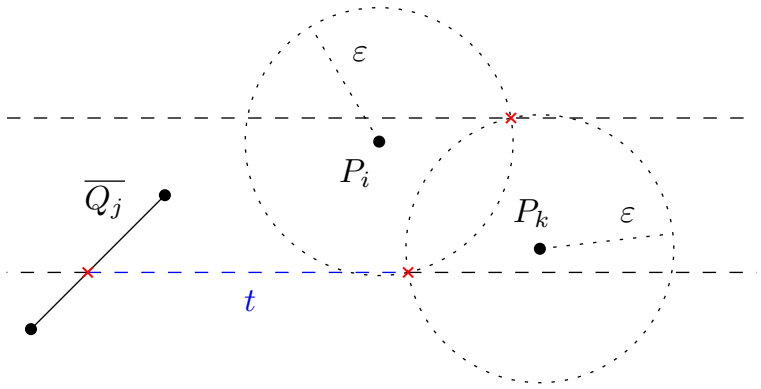


FIGURE 3.7: Example of a critical translation t in an h-configuration.

$(Q_j, Q_k, \overline{P_i})$ is called a v-configuration. A configuration (x, y, s) is either an h-configuration or v-configuration.

Definition 2 (Critical Translations)**.** *Let $c = (x, y, s)$ be an h-configuration and $c' = (x', y', s')$ be a v-configuration. The sets*

$$T^{\varepsilon}_{crit}(c) = \{t \in \mathbb{R} \mid \exists z \in s : d(x, z + t) = d(y, z + t) = \varepsilon\}$$
$$T^{\varepsilon}_{crit}(c') = \{t \in \mathbb{R} \mid \exists z' \in s' : d(x' + t, z') = d(y' + t, z') = \varepsilon\}$$

are called the sets of critical translations for c and c'. A translation is called critical if it is critical for some configuration.

The following two lemma's also come from [8], but these hold similarly for one-dimensional translations.

Lemma 5. *Let $t \in \mathbb{R}$ be a one-dimensional translation. If $\delta_F(P, Q + t) = \varepsilon$, then t is critical.*

Proof. Since $\delta_F(P, Q + t) = \varepsilon$, we know that there must be a monotone path from $(0, 0)$ to (m, n) that is clamped. If the geometric situation corresponds to case 2 (see Figure 2.9), we have the vertices P_i and P_k that both have distance ε to a point $Q(j') + t$ on the line segment $\overline{Q_j} + t$. This means that the h-configuration $(P_i, P_k, \overline{Q_j})$ has t in its critical translation set and therefore t is critical. The same reasoning applies to the geometric situation corresponding to case 1 and for v-configurations. □

It is important to note that there are critical translations t such that $\delta_F(P, Q + t) \neq \varepsilon$, so the condition is necessary, but not sufficient.

Lemma 6. *If there is a translation $t_{\leq} \in \mathbb{R}$ such that $\delta_F(P, Q + t_{\leq}) \leq \varepsilon$, then there is a critical translation $t_{=} \in \mathbb{R}$ such that $\delta_F(P, Q + t_{=}) = \varepsilon$.*

Proof. Let $t_{>} \in \mathbb{R}$ be any translation such that $\delta_F(P, Q + t_{>}) > \varepsilon$. By the continuity property in Lemma 4, it follows that there exists a translation $t_{=}$ between $t_{\leq}$ and $t_{>}$ in the translation space $\mathbb{R}$ such that $\delta_F(P, Q + t_{=}) = \varepsilon$. By Lemma 5, $t_{=}$ must be critical then. □

Consider now the set of all critical translations. Note that it consists of individual points (Figure 3.7) and intervals (Figure 3.8) in the translation space $\mathbb{R}$. From Lemma 6, in order to check if there exists a translation $t_{\leq} \in \mathbb{R}$ such that $\delta_F(P, Q + t_{\leq}) \leq \varepsilon$, it is sufficient to check all critical translations. However, the critical translation intervals that arise from the configurations (x, y, s) where $x = y$ have an infinite number of translations. We call the set of all individual points and endpoints of the intervals the vertices. We prove the following lemma:

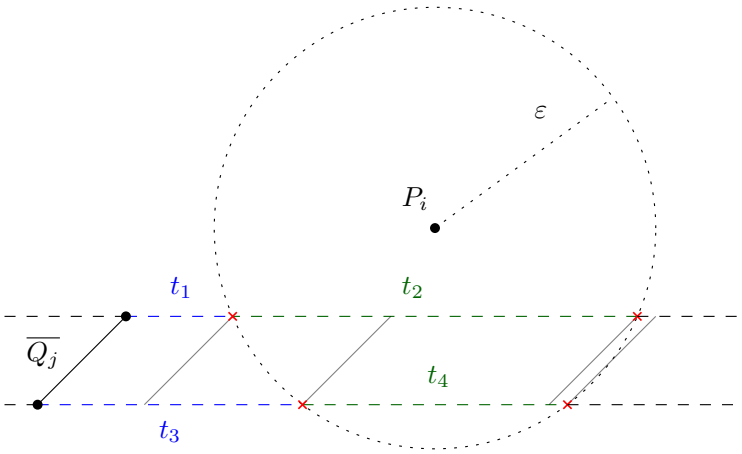


FIGURE 3.8: Example of critical translation intervals $[t_1, t_3]$ and $[t_1 + t_2, t_3 + t_4]$ in an h-configuration (where $P_i = P_k$).

Lemma 7. *For a translation $t_=$ such that $\delta_F(P, Q + t_=) = \varepsilon$ and $t_= \in [t_1, t_2] \subseteq T^{\varepsilon}_{crit}(c)$ for some configuration c, then $t_=$ must be a vertex in the set of all critical translations.*

Proof. If $t_= = t_1$ or $t_= = t_2$, we are done.

Suppose that $c = (P_i, P_i, \overline{Q_j})$ is an h-configuration. Recall that $a^{\varepsilon}_{i,j}$, $b^{\varepsilon}_{i,j}$, $c^{\varepsilon}_{i,j}$, $d^{\varepsilon}_{i,j}$ define the boundaries of the free space. By definition, we have that $a^{\varepsilon}_{i,j} \leq b^{\varepsilon}_{i,j}$ for all translations $t \in T^{\varepsilon}_{crit}(c)$. This means that if there is a monotone path that passes through the single point $a^{\varepsilon}_{i,j} = b^{\varepsilon}_{i,j}$ for $t_=$, then the translations t_1 and t_2 do not close this gap further. A similar argument holds for c being a v-configuration.

Suppose that $t_=$ does not lie on one of the endpoints and once we move away from $t_=$, we have that $\delta_F(P, Q + t_>) > \varepsilon$ for some translation $t_> \in [t_1, t_2]$. Since the gap is not closed, this means that there must be a clamped path from another configuration c' where $t_= \in T^{\varepsilon}_{crit}(c')$. This again means that $t_=$ is either an individual point (in that case we would be done) or lies on some interval of $T^{\varepsilon}_{crit}(c')$, but since $t_> \notin T^{\varepsilon}_{crit}(c')$ as the clamped path got closed, this means that $t_=$ must be an endpoint of that interval. □

From Lemma 7, it follows that it is sufficient to only check the vertices of the critical translations. There are $O(mn(m+n))$ different configurations and each configuration has $O(1)$ critical translations to check: at most two for configurations that involve two different vertices of a curve (Figure 3.7) and at most six for the other configurations (two tangent points of segment to circle and two intersection points for each endpoint with circle; Figure 3.8). Wenk [76] moreover showed a lower bound of $\Omega(mn)$ configurations for this specific case. Each check takes $O(mn)$ time to decide whether $\delta_F(P, Q + t) \leq \varepsilon$ for a critical translation t. It thus takes $O((mn)^2(m+n))$ time in total. This is summarized in the following theorem:

Theorem 3. *For polygonal curves P, Q and $\varepsilon \geq 0$, we can decide whether there exists a one-dimensional translation $t \in \mathbb{R}$ such that $\delta_F(P, Q + t) \leq \varepsilon$ in $O((mn)^2(m+n))$ time.*

For solving the optimization problem, we can use the binary search technique which provides a $(1+\varepsilon)$-approximation in $O(\log(1/\varepsilon))$ searches. For our particular use case, this is a more practical solution than, for example, the parametric search technique with Cole's sorting trick [24].

Theorem 4. *For polygonal curves P, Q and $\varepsilon \geq 0$, we can compute a $(1+\varepsilon)$-approximation of $\delta^*_F(P, Q)$ and its corresponding one-dimensional translation $t^* \in \mathbb{R}$ in $O((mn)^2(m+n)\log(1/\varepsilon))$ time.*

3.3.2 Discrete

For polygonal curves P and Q consisting of m and n segments respectively, we define $d^*_{\mathcal{F}}(P, Q)$ as the minimum discrete Fréchet distance between P and Q over all one-dimensional translations $t \in \mathbb{R}$ of Q:

$$d^*_{\mathcal{F}}(P, Q) = \min_{t \in \mathbb{R}} d_{\mathcal{F}}(P, Q + t)$$

Jiang *et al.* [44] provided a first simple algorithm for matching polygonal curves under translations with respect to the discrete Fréchet distance. For translations restricted to one dimension, the situation becomes more convenient. We introduce a similar algorithm for this. The following observation is important:

Observation 1. *Given two polygonal curves A and B, if there is a translation $t \in \mathbb{R}$ such that $d_{\mathcal{F}}(A, B + t) = \varepsilon$, then there are two vertices $a \in A$ and $b \in B$ such that $d(a, b + t) = \varepsilon$.*

Similarly to the continuous variant, the discrete Fréchet distance also has the continuity property since it is a composite function of the continuous Euclidean distance functions. This implies the following lemma:

Lemma 8. *If there is a translation $t_{\leq} \in \mathbb{R}$ such that $d_{\mathcal{F}}(A, B + t_{\leq}) \leq \varepsilon$, then there is a translation $t_{=} \in \mathbb{R}$ such that $d_{\mathcal{F}}(A, B + t_{=}) = \varepsilon$.*

This means that for a given ε, we can easily determine all the critical translations t such that $d(a, b + t) = \varepsilon$ since each pair $a \in A$ and $b \in B$ has at most two such translations. Therefore, we have a total of $O(mn)$ critical translations and for each translation t, we can check in $O(mn)$ time whether $d_{\mathcal{F}}(A, B + t) \leq \varepsilon$ using the dynamic programming algorithm of Eiter and Mannila [31]. This brings us to the following theorem:

Theorem 5. *For polygonal curves A, B and $\varepsilon \geq 0$, we can decide whether there exists a one-dimensional translation $t \in \mathbb{R}$ such that $d_{\mathcal{F}}(A, B + t) \leq \varepsilon$ in $O(m^2 n^2)$ time.*

Note that this is a factor $O(m + n)$ faster than its continuous variant. For the optimization problem, we can again use the binary search technique which results in the following theorem:

Theorem 6. *For polygonal curves A, B and $\varepsilon \geq 0$, we can compute a $(1 + \varepsilon)$-approximation of $d^*_{\mathcal{F}}(P, Q)$ and its corresponding one-dimensional translation $t^* \in \mathbb{R}$ in $O(m^2 n^2 \log(1/\varepsilon))$ time.*

3.4 Earth Mover's Distance

For two weighted point sets A and B of sizes m and n respectively in $\mathbb{R}^d$, we define $\mathrm{EMD}^*(A, B)$ as the minimum Earth Mover's Distance between A and B over all one-dimensional translations $t \in \mathbb{R}$ of B:

$$\mathrm{EMD}^*(A, B) = \min_{t \in \mathbb{R}} \mathrm{EMD}(A, B + t)$$

We introduce an exact algorithm for the L_1 and L_∞ metrics first and after that, we show a 2-approximation that could provide a solution to other metrics.

3.4.1 L_1 and L_∞

To find a translation t that minimizes $\mathrm{EMD}(A, B + t)$, it suffices to minimize the cost function:

$$D(F, t) = \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d(a_i, b_j + t)$$

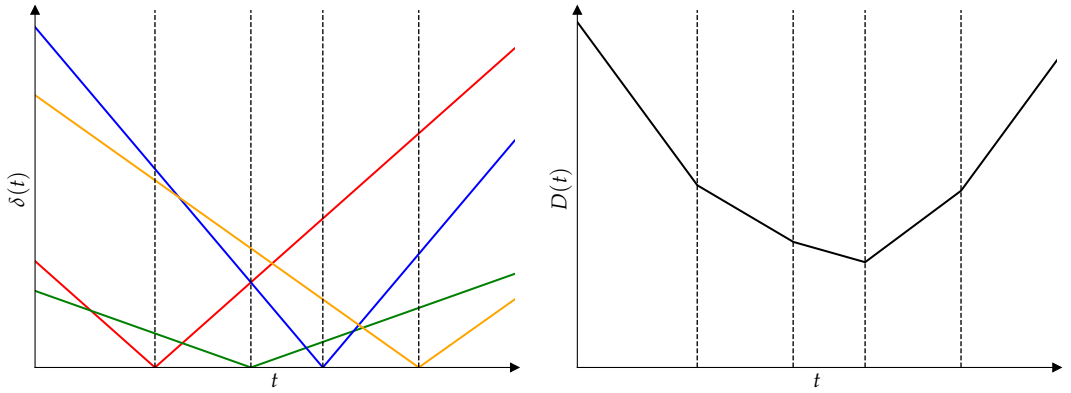
where $F = (f_{ij}) \in \mathbb{R}^{m \times n}$ denotes a flow again. Note that we simply ignore the constant factor that divides the cost by the minimum total weight of A and B. It is not immediately clear how to optimize this function, so let us first analyze the structure of $D(F, t)$ for a fixed (feasible) flow $F = (f_{ij}) \in \mathcal{F}(A, B)$. For a point $a_i \in A$, we denote by a_i^k the k-th component of a_i (similarly for a point $b_j \in B$). For the L_1 metric, the cost function $D(t)$ becomes of the form:

$$\begin{aligned}
D(t) &= \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} \cdot (|a_i^1 - (b_j^1 + t)| + \sum_{k=2}^{d} |a_i^k - b_j^k|) \\
&= \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} \cdot (|x_{ij} - t| + c_{ij}) \\
&= \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} |x_{ij} - t| + f_{ij} c_{ij} \\
&= \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} |x_{ij} - t| + \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} c_{ij} \\
&= \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} |x_{ij} - t| + C
\end{aligned}$$

where $x_{ij} = a_i^1 - b_j^1$, $c_{ij} = \sum_{k=2}^{d} |a_i^k - b_j^k|$ and $C = \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} c_{ij}$ are constants. For simplicity, we will again ignore the constant factor C. Note that each function $\delta_{ij}(t) = f_{ij}|x_{ij} - t|$ is a piecewise linear function, so the cost function $D(t)$ is the sum of these piecewise linear functions, and therefore, also a piecewise linear function itself. We observe that the slope and intercept of the piecewise linear function $D(t)$ changes each time a function $\delta_{ij}(t)$ changes its slope and intercept, which happens at $t = x_{ij}$ where the course of the function switches from $f_{ij}(x_{ij} - t)$ to $f_{ij}(-x_{ij} + t)$. Since $f_{ij} \geq 0$, this means that $D(t)$ is first gradually decreasing and at a certain point x_{ij} gradually increases again, which means that the minimum also lies at that point. An illustration of this is given in Figure 3.9. This means that, for a fixed flow F, we can compute the corresponding optimal one-dimensional translation using a simple plane sweep from left to right in $O(mn \log(mn))$ time (since sorting the mn event points dominates the runtime).

Let us now consider the L_∞ metric. The cost function becomes of the form:

$$\begin{aligned}
D(t) &= \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} \cdot \max\{|a_i^1 - (b_j^1 + t)|, \max_{k=2}^{d} |a_i^k - b_j^k|\} \\
&= \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} \cdot \max\{|x_{ij} - t|, c_{ij}\}
\end{aligned}$$



(A) Piecewise linear functions $\delta(t)$ (B) The cost function $D(t)$

FIGURE 3.9: Example of the cost function structure for the L_1 metric.

where $x_{ij} = a_i^1 - b_j^1$ and $c_{ij} = \max_{k=2}^{d} |a_i^k - b_j^k|$ are constants again. Note that we again obtain a piecewise linear function composed of the functions $\delta_{ij}(t) = f_{ij} \cdot \max\{|x_{ij} - t|, c_{ij}\}$. Each function $\delta_{ij}(t)$ is also composed of the piecewise linear functions:

$$\delta_{ij}(t) = \begin{cases} f_{ij}(x_{ij} - t) & \text{if } t \leq x_{ij} - c_{ij} \\ f_{ij}c_{ij} & \text{if } x_{ij} - c_{ij} < t \leq x_{ij} + c_{ij} \\ f_{ij}(-x_{ij} + t) & \text{if } t > x_{ij} + c_{ij} \end{cases}$$

We can therefore use a similar approach that we used for the L_1 metric, but each function $\delta_{ij}(t)$ now results in (at most) two event points $t_1 = x_{ij} - c_{ij}$ and $t_2 = x_{ij} + c_{ij}$ where the intercept and slope of $D(t)$ changes. The procedure is besides that identical.

So we know now how to optimize this distance function with respect to a fixed feasible flow, but our goal is to find the global optimum for any flow. We observe that for any flow F, we always have the same set of event points (that is, the event points are independent of the flow). This means that we can simply compute for each such event point t the corresponding optimal flow by static computation of $\text{EMD}(A, B + t)$. The event point t^* that corresponds to the lowest distance is then the optimal one-dimensional translation for any flow. Let $T(m, n)$ be the time it takes to compute the Earth Mover's Distance between two sets of points of sizes m and n (using any algorithm). We obtain the following theorem:

Theorem 7. *Given two weighted point sets A and B of sizes m and n, we can compute the optimal one-dimensional translation in $O(mn \cdot T(m, n))$ time for the L_1 and L_∞ metrics.*

3.4.2 2-approximation

For translations in arbitrary directions, it has been shown that exact computation in case of the L_2 metric is not possible due to the reduction from the geometric median [19]. Although our problem is a bit more restricted, we still expect that exact computation is not possible. Therefore, for this particular case, we have to rely on approximations. Cabello *et al.* [20] showed a 2-approximation when we consider the translations that match each pair of points and take the minimum Earth Mover's Distance of those. We show that this also holds for matching points only along one dimension. Note that in case of the L_1 metric, this provides an exact solution as shown earlier.

Given two weighted point sets A and B, let t_{ij} denote the translation that matches the points $a_i \in A$ and $b_j \in B$ on the x-axis and let t^* denote an optimal translation. We have the following theorem:

Theorem 8. $\min_{a_i \in A, b_j \in B} EMD(A, B + t_{ij}) \leq 2 \cdot EMD(A, B + t^*)$ *if the underlying metric is any L_p-norm with $p \geq 1$.*

Proof. It is obvious that there exists a pair of points $a_{i^*} \in A$ and $b_{j^*} \in B$ such that

$$d(a_{i^*}, b_{j^*} + t^*) \leq d(a_i, b_j + t^*)$$

for any $a_i \in A$ and $b_j \in B$. For a point $b_j \in B$, the distance between the points $b_j + t^*$ and $b_j + t_{i^*j^*}$ is simply the difference in their translations, i.e.,

$$d(b_j + t^*, b_j + t_{i^*j^*}) = |t^* - t_{i^*j^*}|$$

This is exactly the distance along the x-axis between a_{i^*} and $b_{j^*} + t^*$, and since d is any L_p-norm with $p \geq 1$, it follows that

$$\begin{aligned} d(b_j + t^*, b_j + t_{i^*j^*}) &\leq d(a_{i^*}, b_{j^*} + t^*) \\ &\leq d(a_i, b_j + t^*) \end{aligned}$$

By triangle inequality, we have

$$\begin{aligned} d(a_i, b_j + t_{i^*j^*}) &\leq d(a_i, b_j + t^*) + d(b_j + t^*, b_j + t_{i^*j^*}) \\ &\leq 2 \cdot d(a_i, b_j + t^*) \end{aligned}$$

Hence, we have

$$\begin{aligned} \min_{a_i \in A, b_j \in B} \mathrm{EMD}(A, B + t_{ij}) &\leq EMD(A, B + t_{i^*j^*}) \\ &= \min_{F \in \mathcal{F}(A,B)} \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d(a_i, b_j + t_{i^*j^*})}{\min(W^A, W^B)} \\ &\leq \min_{F \in \mathcal{F}(A,B)} \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} 2d(a_i, b_j + t^*)}{\min(W^A, W^B)} \\ &= 2 \cdot EMD(A, B + t^*) \end{aligned}$$

□

Chapter 4

Methodology

In this chapter, we describe the methodology of this research. We start with a formal description of the problems that are related to hyperbolic localization, which includes the TDOA estimation and show how this connects to the matching under one-dimensional translations problem that we discussed in Chapter 3. We then describe the datasets we are using and specify the pre- and post-processing that needs to be done. Finally, we describe the general experimental setup and evaluation methods.

4.1 Hyperbolic Localization

In Chapter 2, we already provided a high-level overview of the concepts related to hyperbolic localization. This section builds on that and goes into more detail. Recall that acoustic localization using hyperbolic localization methods is accomplished in two stages. The first stage involves the estimation of the TDOAs between the microphones, where we take a geometric approach to achieve this. In the second stage, we try to produce an unambiguous solution to the nonlinear hyperbolic equations that arise from these TDOAs using efficient techniques.

4.1.1 Geometric TDOA Estimation

Problem Statement. For a signal of interest $s(t)$, the general model for the time delay estimation between signals received at two different recorders, $x_1(t)$ and $x_2(t)$, is given by:

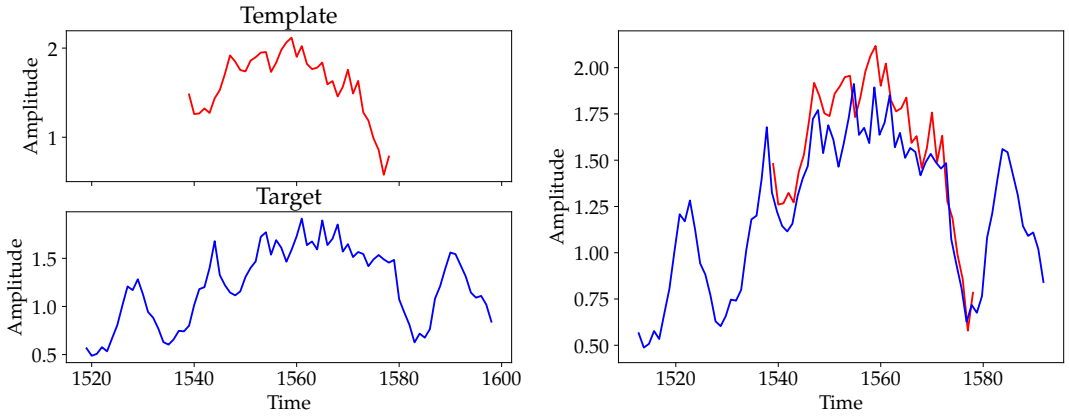
$$x_1(t) = a_1 \cdot s(t - \tau_1) + n_1(t)$$
$$x_2(t) = a_2 \cdot s(t - \tau_2) + n_2(t)$$

where a_1 and a_2 are the amplitude scaling which represent the attenuation that is experienced during propagation of the signal, τ_1 and τ_2 represent the time delay of the signal from the source to the recorders, and $n_1(t)$ and $n_2(t)$ are the additive noise. This model can be simplified further if we take first recorder as reference:

$$x_1(t) = s(t) + n_1(t)$$
$$x_2(t) = a \cdot s(t - T) + n_2(t)$$

where $a = \dfrac{a_2}{a_1}$ is the ratio of the amplitude scaling factors and $T = \tau_2 - \tau_1$ is the difference in time between the signals. Our goal is to estimate T.

We can turn this into some kind of optimization problem. Informally, let x_1 and x_2 be the signals, either in their waveform or spectrogram representation, and let D be some distance function. We then try to find a time shift t^* such that $D(x_1, x_2 + t^*)$



(A) Initial setting (B) Solved setting

FIGURE 4.1: Example of a partial matching between time series data.

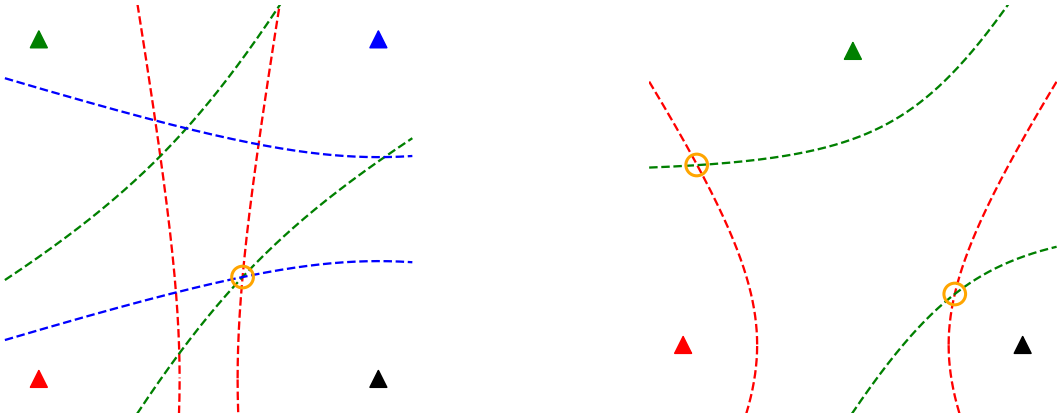
is the minimum value across all translations. This is exactly how the cross-correlation methods operate, where they calculate a similarity value at each sliding step and takes the offset with the largest similarity value. If we take a geometric approach instead, the waveform and spectrogram representations of the signals are treated as two-dimensional curves and three-dimensional surfaces in space and D represents some geometric distance measure. Below we provide a formal description of the problem. Although sound is typically captured in a discrete form, we will treat them as continuous beings.

Geometric Approach. Let $f(x) : \mathbb{R}_{\geq 0}^{d-1} \to \mathbb{R}$ denote a function that returns the amplitude for each sample $x \in \mathbb{R}_{\geq 0}^{d-1}$ in dimension $d = 2, 3$. We denote the set

$$A = \{(x, f(x)) \mid x \in \mathbb{R}_{\geq 0}^{d-1}\} \subseteq \mathbb{R}^d$$

as the graph of the function. Note that for $d = 2$, this is the two-dimensional x-monotone curve, and for $d = 3$, this is the three-dimensional surface (more specifically, a terrain). Let $A, B \subseteq \mathbb{R}^d$ be the curves or surfaces for $d = 2$ and $d = 3$ respectively that represent the graphs of the functions f and g corresponding to their signals. The x-component always represents the time. For $d = 2$, the y-component represents the amplitude. For $d = 3$, the y-component represents the frequency and the z-component represents the amplitude. From a set of possible shifts in time $\mathcal{T} \subseteq \mathbb{R}$, we would like to find the optimal shift $t^* \in \mathcal{T}$ such that the distance between the shapes is minimized. This is exactly the problem of matching under one-dimensional translations.

However, this assumes that A and B represent the same segment of the signal, which would imply that we already have information on the time delay between them. This therefore leans more towards a partial matching problem, where A is a (continuous) template signal that we are trying to optimally match against part of a (continuous) target signal B of longer duration. Suppose that $B^* \subseteq B$ is the part of the shape that corresponds to the same segment A, we can safely assume that (1) B^* must have the same duration as A, and (2) B^* must also be continuous (i.e., no interruptions). If the template has a duration of $dur(A)$ along the time axis, we can define a set of partial shapes $\mathcal{B} \subseteq \mathcal{P}(B)$ that should satisfy the constraint of having



(A) Single location (B) Ambiguous location

FIGURE 4.2: Hyperbolas defined by the difference in distance compared to a reference. The triangles denote the microphones with the black triangle being the reference.

equal duration and being continuous (*cont*). More formally:

$$\mathcal{B} = \{B' \in \mathcal{P}(B) \mid dur(B') = dur(A) \wedge cont(B')\}$$

Note that for the discrete case, both signals are sampled at the same fixed sample rate, which means that the set $\mathcal{B}$ simply consists of all continuous segments of B with an equal number of samples to A. This allows us to match A on each partial shape $B' \in \mathcal{B}$ independently and find the best matching with its corresponding time shift, which is again exactly the problem of partial matching under one-dimensional translations. We can therefore use the algorithms described in Chapter 3 to solve the TDOA estimation. An illustration of the problem is shown in Figure 4.1.

4.1.2 Localization Technique

Problem Statement. For an array consisting of M microphones, we assume that the TDOAs are referred to the first microphone, which is often the microphone that is closest to the source. Let (x, y) be the (unknown) source location and (X_i, Y_i) be the (known) location of the i-th microphone for $i = 1, ..., M$. The (Euclidean) distance between the source and the i-th microphone is given by:

$$R_i = \sqrt{(X_i - x)^2 + (Y_i - y)^2}$$

The difference in the distance between the first microphone and i-th microphone is then given by:

$$\begin{aligned} R_{i,1} &= c \cdot \tau_{i,1} = R_i - R_1 \\ &= \sqrt{(X_i - x)^2 + (Y_i - y)^2} - \sqrt{(X_1 - x)^2 + (Y_1 - y)^2} \end{aligned}$$

where c is the (known) propagation speed and $\tau_{i,1}$ is the (known) estimated TDOA between the first microphone and i-th microphone. This defines a set of non-linear hyperbolic equations whose solution gives the two-dimensional location of the source.

Ideally, the system of equations provides a unique solution. An example of this is shown in Figure 4.2a where we have four microphones and the closest microphone

to the source is considered the reference microphone. The hyperbolas defined by each microphone intersect each other at a common point, which is the source location. However, there are several reasons that could lead to this system of equations to be inconsistent (e.g., inaccurate TDOA measurements or microphone locations) where the intersection of the hyperbolas does not coincide. In that case, it is required to select an optimum solution using some error criteria. Note that if the set of equations equals the number of unknown coordinates of the source location (e.g., two hyperbolas for a two-dimensional position), then the system is by definition consistent as there is always a unique solution. In practice, we often have more equations to minimize the error.

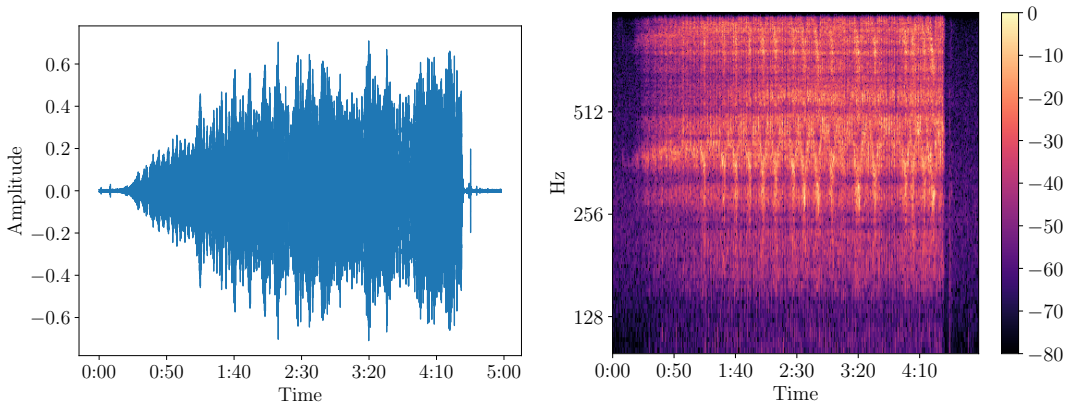
Techniques. Solving these non-linear hyperbolic equations is difficult. Most of the times, the equations are linearized to simplify the computation, which generally does not introduce errors in the estimation. However, in situations where the microphones are poorly placed with respect to the source location (known as dilution of precision), it can introduce significant errors [12]. The hyperbolic localization algorithms are classified into two categories: iterative and non-iterative [33], which both have their advantages and disadvantages. Iterative algorithms do not always converge due to the need of a proper initialization setup, and non-iterative algorithms always yield two feasible solutions (a positive and negative root) of which one is closer to the actual source location, but it is not immediately clear which of the two. An example of the output locations from the two roots is shown in Figure 4.2b for a setup with 3 microphones.

In this thesis, we will be using the least-squares method as developed by Stephen Bancroft [12], which is a non-iterative algorithm originally designed for GPS systems, but it also applicable to various other localization systems including TDOA localization. It provides a closed-form algebraic solution to the estimated position that minimizes the sum of squared residuals between the observed and predicted TDOAs. A brief description of the mathematical formulation can be found in Appendix A. Note that all the given formulations can easily be extended to three dimensions, but in this thesis, we are more interested in finding an accurate two-dimensional position.

Sound Finder. The algorithm of Bancroft requires the specification of a speed of sound and since it returns two root solutions, one must be selected as output. The Sound Finder software [78] is an implementation of this and makes concrete decisions on these aspects. The speed of sound c (m/s) is determined by:

$$c = 331.3 \cdot \sqrt{1 + \frac{T}{273.15}}$$

where T is the temperature in °C. Moreover, the root with the lower sum of squares discrepancy is selected as output estimate position. In Figure 4.2a, this would mean that we obtain the root location where all hyperbolas perfectly intersect since the discrepancy is zero, whereas the other root location clearly has a higher discrepancy. This also shows why at least four microphones is required to unambiguously determine the position, since the discrepancy of the two root solutions in the setup with three microphones are both zero (see Figure 4.2b again).



(A) Waveform (B) Spectrogram

FIGURE 4.3: High-quality roar (after preprocessing).

4.2 Data

In this section, we describe the three data sources that we will be using, which are:

- **Real:** roars obtained from the field that we aim to accurately localize.
- **Simulated:** synthetically construct more roars to evaluate localization (and TDOA estimation) performance in a controlled simulation environment.
- **Synthetic Signals:** generation and modification of signals to evaluate matching or TDOA estimation performance.

4.2.1 Real

This study uses a dataset of the Guianan red howler monkey that is provided by Yannick Wiegers, Julia Blok, Martijn Ruisbroek, Tom Simmelink, and Georgis Rallis, who are affiliated with Utrecht University.

During the months of November and December of 2024, five arrays consisting of either five or 10 microphones have been deployed at known howler monkey territories in French Guiana. Close to the arrays, a total of eight roars were emitted at dawn and have been sufficiently captured by at least four microphones of a single array. The two-dimensional source locations have been measured, together with the three-dimensional locations of the microphones themselves (see Figure 4.4). The microphones recorded in single-channel WAV format for the duration of ± 5 minutes with a sampling rate of 24 kHz (see Figure 4.3).

Despite the fact that the Guianan red howler monkey is currently listed as "Least Concern", this research mainly aims to study the feasibility of localizing primates using our methods and most of the results obtained from this dataset likely generalize to other howler monkey species as well (or even other primates) due to similarities in the characteristics of their vocalizations. As they are still hunted locally, the localization of this specific species could help in revealing potential danger early for effective conservation.

4.2.2 Simulated

For a large-scale evaluation of our methods on the localization of howler monkeys, a simulation of roars is required. In a simulation setup, we have a virtual environment

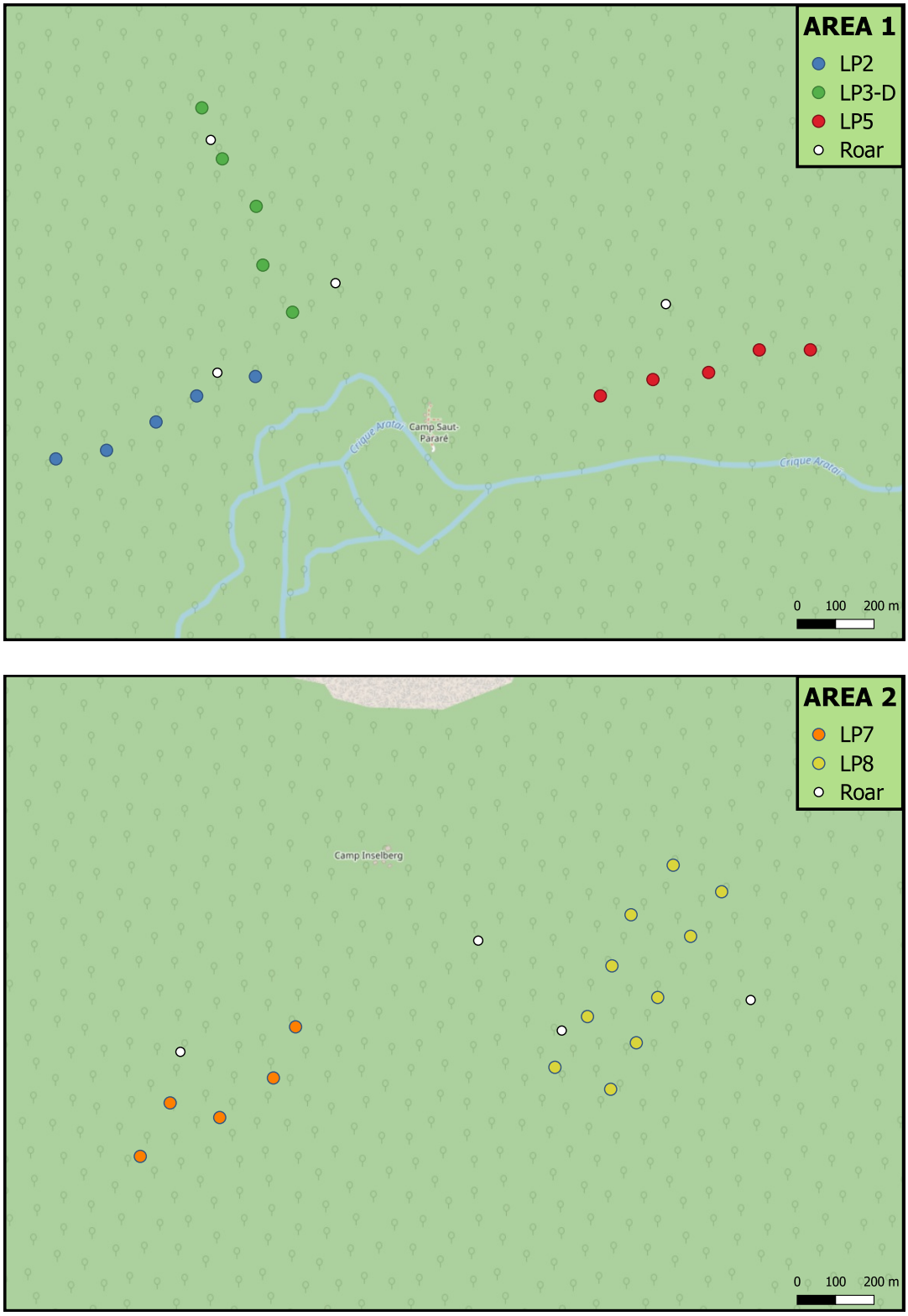


FIGURE 4.4: Microphone setup and roar locations in two different areas (Réserve naturelle des Nouragues, French Guiana).

where we can manually specify microphone and source locations. The idea is then to play a recording at a source location and obtain a modified recording as if it was captured by the respective microphone in this environment. We simulate a forest environment of 1×1 km with a speed of sound that matches with the localization algorithm. Since we focus on two-dimensional localization, we set for all objects (i.e., microphone and source locations) an equal height of 1.5 meters (which will the be irrelevant). The microphones will furthermore only be placed in the range $[200, 800] \times [200, 800]$ (in meters) for two reasons: to prevent edge-effects and ensure that the TDOAs will not exceed the limit set by our matching algorithms.

To simulate recordings in a forest environment, we can use a forest impulse response (IR) simulation algorithm. An impulse response describes how a system reacts to an impulse, which is a very short and instantaneous input signal. In acoustics, this system describes a certain environment, such as a concert hall, where an impulse response represents how sound propagates from a source to a receiver in that specific environment. If we convolve an impulse response with a sound signal, we can simulate how that signal would sound in the modeled environment. Formally, if we have an input signal $x[t]$ and an impulse response $h[t]$, the convolved signal $y[t]$ is obtained by:

$$y[t] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[t-k]$$

So for each source-microphone pair in the simulation, we construct such an impulse response and convolve this with a (high-quality) roar to obtain the simulated recording.

Kaneko and Gamper [45] fairly recently introduced an algorithm for this that models acoustic scattering in a forest caused by tree trunks using single scattering cylinders. The same authors later employ this algorithm for simulating forest scenes to examine the performance of distributed microphone arrays for bird localization [46], but it has not yet been employed on other use cases. We use this for our simulation setup. We set the number of trees to zero, since we are mainly interested in the performance of our methods in a best-case scenario. Besides, the reverberation effect caused by the trees alone did not seem to have a significant impact on our roar signal, so the performance of the TDOA estimation is not expected to significantly differ either. For a more realistic simulation, one would additionally need separate ambient (or background) noise recordings that are appropriate for the aimed environment. The simulation returns IRs of 5-second duration with a sample rate that matches that of the roar recordings, allowing for the convolution.

4.2.3 Synthetic Signals

Our goal is to develop methods that are more robust to noise and attenuation effects, which are common in real-world signals, for more accurate estimation of the TDOA. This trivially leads to more accurate localization. To make reliable statements about the robustness characteristics of the methods, we sometimes have to rely on synthetic signals since we have limited information on the quality of the real dataset. For example, we can not determine an approximate SNR of each recording since they do not contain sufficient samples of isolated background noise. Other than that, synthetic data allows for a larger-scale evaluation where we have full control over relevant parameters that simulate these effects. For the signal type, we either consider fully synthetic signals (e.g., sine wave) or a (high-quality) roar from the real dataset. We also need to specify a noise model and time shift strategy.

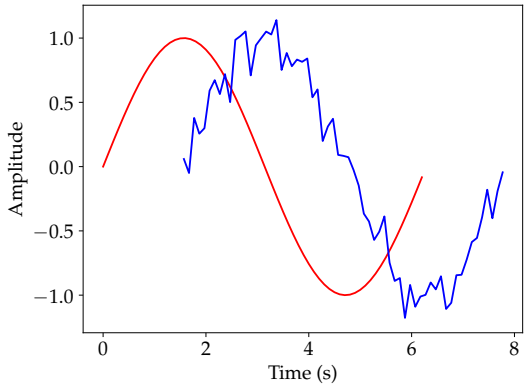


FIGURE 4.5: Example of a noisy shifted signal (SNR=15dB and relative shift is 25%).

Noise Model. The additive white Gaussian noise (AWGN) is often used in signal processing to mimic the effect of many random processes that occur in nature. Noise is modeled as a Gaussian distribution with zero mean and a specified standard deviation σ, which corresponds to a certain SNR level.

Given the power of the signal P_{signal} and a desired SNR level, we can determine the power of the noise P_{noise} (rewrite of the equation given in Chapter 2), which comes down to the variance of the noise σ^2. To additionally simulate the effect of attenuation, we multiply the original signal with an attenuation factor $\alpha \in [0, 1]$. In practice, if we have a clean signal $y[t]$ and a sampled noise $n[t] \sim \mathcal{N}(0, \sigma^2)$, the noisy signal simply becomes

$$y'[t] = \alpha \cdot y[t] + n[t]$$

Time Shift. For the evaluation of the matching or TDOA estimation performance, it is desired to also add a time shift to the noisy signal. The strategy of determining a time shift depends specifically on the matching setting. In case we perform a regular matching, the shift is relative to the original signal and will be uniformly sampled from $[-1, 1]$ (e.g., 0.25 represents a shift of 25% with respect to the original shape, see also Figure 4.5). Note that for most algorithms, the initial position is irrelevant to the matching position and only affects the translation. For the partial matching setting, the shift is absolute and will be uniformly sampled within a range depending on the TDOA limit set by our matching algorithms (e.g., if we allow TDOAs of at most 2.5 seconds, we take the range $[-2.5, 2.5]$).

4.3 Pre- and Post-processing

In this section, we detail three preprocessing steps that are applied to the signal data and a postprocessing step that is required to correct for synchronization.

Downsampling and Band-limiting. The dominant frequency band of the howler monkey roars are in the 100-1000 Hz range. By the Nyquist-Shannon sampling theorem, we need a sampling rate of at least 2 kHz to prevent aliasing. The recordings are therefore downsampled to 2 kHz. Additionally, a band-pass filter of 10-1000 Hz is applied to the data to filter out the irrelevant frequencies.

Noise Removal. Since the recordings contain errors with segments of complete silence that affect the TDOA calculation, we have to filter them out. Empirically determined, we look at reach region of 128 samples with an overlap rate of 50% and compare its loudness to a threshold below the peak of 60 dB. This means that any parts that are up to 60 dB quieter than the peak are considered non-silent and may stay, whereas the other parts have a silent level that is even uncommon to regular background noise and are therefore removed.

Normalization. Since we want to compare the shapes of the signals rather than their absolute magnitudes, we have to normalize the data. This ensures that the amplitude differences do not bias the comparison. One that is often used in signal processing is peak normalization, which scales the data into the fixed range $[-1, 1]$ based on the largest peak. However, this is sensitive to outliers as the overall energy distribution might be significantly different. In the case of real-world signal data, the presence of outliers is very likely. To better match the energy levels of the signals, root mean square (RMS) normalization is a more suitable alternative. For a set of values X of size N, the normalized set X' is then obtained as follows:

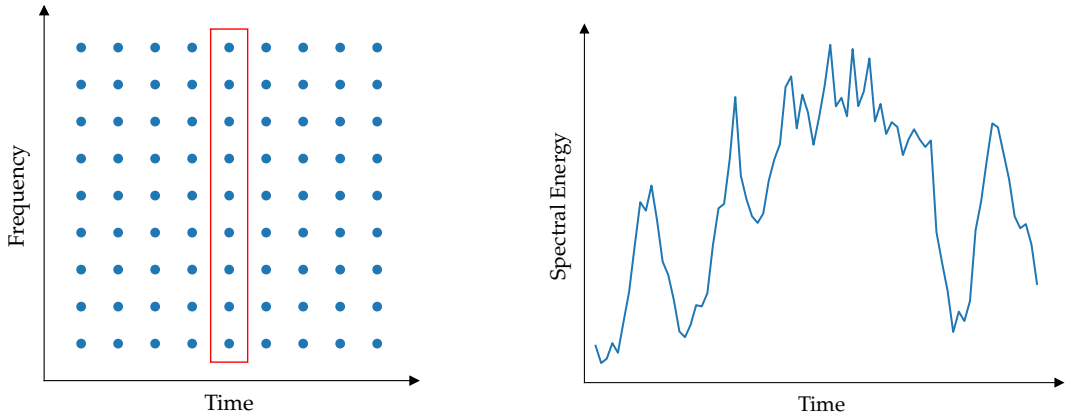
$$X' = \frac{X}{RMS(X)} \quad \text{where } RMS(X) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} X_i^2}$$

where X_i represents the i-th value in X. Moreover, the RMS value is a good indication of the resolution of the signal, which we can use to determine which microphones best captured the signal. Often times, these are the microphones closest to the source location, on which we obviously have no information in practice.

Synchronization. To synchronize the recorders, we apply the acoustic synchronization method by playing a beep at the start and end of deployment, when the recorders are together (such that they capture this beep simultaneously). This beep is captured at different timestamps by each recorder, so these are first identified and relative differences are taken between the recorders. We then obtain offsets for both the start and end points. Calls that occurred within 24 hours of one of these moments are synchronized using their respective offsets by subtracting them from the TDOA estimates. Outside this range, we need to correct for drift on top of the start synchronization (so we assume drift to be negligible within this range). If we assume that drift is linear, we can calculate a drift rate based on these measurements again. For example, if we have two recorders where the two beeps were captured 600 seconds apart for the first recorder and 600.3 seconds apart for the second recorder, we then obtain a drift rate of $0.3/600 = 0.5$ ms/s for the second recorder. After 10 seconds, its internal clock is 5 ms ahead compared to the first recorder. This must also be subtracted from the TDOA estimates.

4.4 Experimental Setup

This section describes the general experimental setup, which involves the transformation of the signal data to a suitable representation, strategies for selecting template segments to match on, parameter settings, and the technical specifications.



(A) Grid of (weighted) points

(B) Spectral energy envelope

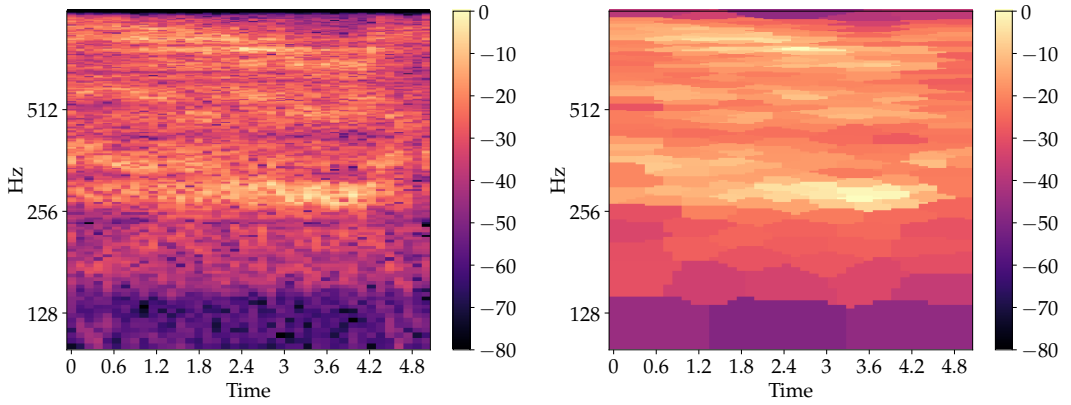
FIGURE 4.6: Examples of signal representations. The spectral energy envelope is obtained by aggregating the frequency bins at each time bin.

4.4.1 Signal Representation

The waveform of the signal is a rich representation, containing both the phase and amplitude information of the entire signal. It has a high temporal resolution with a sample rate of 2000 Hz, meaning that at each $1/2000 = 0.0005$ second, we have a sample of the original signal. While this gives enough information for reconstructing the original signal, using the waveform directly as input of our methods will be infeasible (e.g., a 5-second template requires 10.000 points). We can not downsample the signal further as we would lose important frequency information about the roars. Taking a shorter segment as template may be insufficient for a proper matching and does not fully resolve the issue as it still requires a considerable amount of matchings with the partial shapes. Our focus will be on the spectrogram representation, which is more compact as it considers segments of the signal in time and also discards the phase information.

Spectral Energy Envelope. While this representation already reduces the complexity of the original waveform, it still does not significantly reduce the number of points due to its extra dimension. Note that it consists of a set of points in a grid with weights associated to them, representing the amplitude values at each time and frequency point. We need to reduce this grid to a fewer number of points that still effectively approximates the original grid. We can aggregate the frequency bins at each time bin to reduce this three-dimensional spectrogram to a two-dimensional energy envelope of the signal over time (see Figure 4.6). Not only is this a more compact representation than the waveform, but it is also less oscillatory, making it easier to observe patterns in the shape, which consequently should improve the matching as well (Figure 4.1 shows an example of such a matching). On top of that, it comes with the benefit of having non-negative values only, meaning that we can directly model them as weights in the Earth Mover's Distance for example. This representation will mainly be our focus.

Spectrogram Approximation. The main disadvantage of the spectral energy envelope is that we still lose frequency-specific information, which could potentially be useful in the matching process for more accurate TDOA estimation. Therefore, it



(A) Normal (9240 vertices) (B) Approximated (300 clusters)

FIGURE 4.7: Example of a weighted K-means approximation with averaging weights of points in same cluster. Other approximations can be found in Figure B.6 of Appendix B.

would be interesting to also observe how the algorithms respond if we retain this extra dimension. We could perform a similar procedure as previously where we aggregate bins in a small $L \times W$ window for some $L, W \in \mathbb{N}$ and slide it over the entire grid, but we can also do something more targeted knowing that we are more interested in the regions with high amplitude values where the roar is more pronounced. Namely, the lower amplitude regions tend to fluctuate more due to noise (or other factors that, for example, involve the STFT transformation), which makes them less suitable to match on. We will consider two options for this: weighted K-means [60], which is a clustering-based approach, and a greedy insertion approach [34], which samples important points. A brief description of these methods can also be found in Appendix A. The amplitude of the cluster centers obtained from the weighted K-means is determined by aggregating (sum, max, or average) the amplitudes of the points that belong to the same cluster. An example of a K-means approximation is shown in Figure 4.7.

4.4.2 Template Selection

Throughout the experiments, we sometimes need to choose which recording of a roar is used as template (or target). In most cases, we simply take the recordings with the highest resolutions as determined by the RMS value. We assume that these best represent a clean roar (without significant background noise) and results obtained on them likely generalize well to other high-quality roar recordings.

Strategy. Since we only consider a small segment of the entire signal as template, it is important to select a segment with a distinct shape that can be matched unambiguously to the target segment. Taking a random segment of the signal could be one strategy, however, this might select segments of background noise that can not be properly matched due to their fluctuating patterns. Similarly as with the spectrogram approximations, we should focus on the parts where the roar is most pronounced. These are simply the peaks in the spectral energy envelope. Suppose we have a peak at time bin x and we have a template length of L (which we expect to be even), we then center the template around x such that the template range becomes $[x - \frac{L}{2}, x + \frac{L}{2}]$ (and correct for boundaries by shifting the center position if necessary).

Top-k. In the experiments, we either focus on a single peak or a top-k of peaks. If we are using a top-k, we want the templates to be sufficiently different from each other (e.g., a largest peak at time bin x and a second largest peak at $x + 1$ is generally not different enough). Therefore, the time bins that were part of the template in the prior peaks are excluded from the candidate list. This concretely means that, with our strategy, the maximum overlap between the templates is 50%.

4.4.3 Parameters

We list the parameters used throughout the experiments below. Note that the focus here is on the general setting from which we do not deviate unless otherwise specified. Other parameters are explicitly stated with their experimental results.

STFT. For the STFT procedure to transform the waveform to a spectrogram representation, we choose a FFT size (n_fft) and window size (win_length) of 512 samples. Recall that the window size determines the length of the segment that is used for each Fourier Transform. The FFT size is a different parameter where we can control the number of frequency bins without affecting the time resolution, so setting a larger FFT size than the window size would require zero-padding (which does not add any new information to the signal). The hop size (hop_length) is set at 256 samples, resulting in an overlap of 50% between consecutive STFT frames. The window function ($window$) is Hanning, which reduces spectral leakage more effective than other functions. Since our data is sampled at a sample rate of $f_s = 2000$ Hz, these settings correspond to a frequency resolution of $f_s/n\_fft = 2000/512 = 3.91$ Hz and time resolution of $hop\_length/f_s = 256/2000 = 0.128$ seconds. Since the frequencies between 0 and 100 Hz are filtered out with a band-pass filter, the bins corresponding to these frequencies are removed from the spectrogram which leaves us with a total of 231 frequency bins.

Template and Target Lengths. Taking into account the computational restrictions that we are dealing with, we set a template length of 40, corresponding to a segment of $40 \cdot 0.128 = 5.12$ seconds (the effectiveness will be experimentally validated as well). For the target length, we assume that the TDOA is at most 2.5 seconds. If we take the target segment at the same timestamp as the template segment, this means that we need to add another $2.5/0.128 = 20$ samples before and after. In reality, this means that if we draw a line between two microphones and the source location lies on this line, but not on the interior of the segment, then the microphones should have a distance of at most $2.5 \cdot 346.71 = 866.77$ meters. While this is not always the case for certain setups, the signal strength has likely weakened at such distances to a point where TDOA estimation becomes very difficult. In most setups, there are enough microphones closer to the source that can be used for the TDOA estimation instead. Moreover, our data matches this assumption, so setting a higher upper bound would only be redundant.

Temperature. For the localization, we need to specify a temperature which determines the speed of sound. Since this was not measured during the recordings of the roars, a fixed temperature of $T = 26^\circ$C is assumed. Using the formula from the Sound Finder software, this corresponds to a speed of sound of roughly 346.71 m/s.

Binary Search. For the Fréchet distance matching algorithms, we need to specify a range for the binary search, which terminates when the difference between the higher

and lower bound is below a specified ε. We set an initial lower bound of zero and a greedy upper bound that is determined by the maximum of the two static distances between the curves if we match the last vertex of one curve with the first vertex of the other curve. This is clearly a valid upper bound since moving the curves further away from each other can only increase the Fréchet distance.

4.4.4 Technical Specifications

In this section, we describe the implementation details. Note that if we refer to any library, we mean the Python version of this, unless otherwise specified.

Matching Algorithms. The shape matching algorithms have been implemented in Python 3.10.7, with the aim of making the algorithms easily usable for people without a programming background as well. The code is available at our GitHub repository [75], accompanied with instructions on how to run the code.

For most of the array processing, we used the *numpy* library [38]. For static computation of the discrete Fréchet distance, we used the *frechetdist* library [29], which is an implementation of the dynamic programming algorithm of Eiter and Mannila [31]. We used a custom implementation for the continuous Fréchet distance due to a lack of available libraries, based on the algorithm of Alt and Godau [6]. For static computation of the Earth Mover's Distance, we used the *scipy* library [74], which provides an efficient implementation for both the one-dimensional case through the function *wasserstein_distance*, and for the higher-dimensional case through *wasserstein_distance_nd* (that uses the Euclidean distance as underlying metric).

One additional optimization we made to the implementation of the algorithms exploits the observation that the point sets of our data are sampled at a constant sampling rate, which means that they are evenly spaced (i.e., the distance between two consecutive points on the x-axis is always the same). For the Earth Mover's Distance, instead of a quadratic number of translations, we would only have to consider a linear number of translations.

Other Libraries. For loading and processing the audio files, we used the *librosa* library [54]. The Sound Finder [78] implementation in R is used to perform the localization, which supports both two- and three-dimensional localization. The cross-correlation TDOAs are calculated using the *numpy* library again. For the spectrogram approximations, we used the *scikit-learn* library [60] to obtain the clusters from the weighted K-means. For the terrain approximation using greedy insertion as described by Garland and Heckbert [34], we took the *pydelatin* library [13].

Hardware. All experiments are run on a Windows 11 notebook with an Intel Core i7-8750H 2.20GHz 6-cores processor and 16GB RAM.

4.5 Evaluation

The evaluation consists of three main parts: the errors made on the matchings and TDOAs, the errors of the estimated positions and the quality of the microphone setup.

Matching Error. For measuring the translation error between a single matching, we simply take the absolute difference between the estimated x and true shift y (i.e.,

$e = |y - x|$). If we have a set of n samples, we can take the mean absolute error (MAE) defined as:

$$\text{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} e_i}{n}$$

The main advantages of this metric are its interpretability and robustness to outliers.

TDOA Error. For measuring the error of the TDOA, we could again use the MAE. However, this gives equal weight to all errors, but a larger error in the TDOA estimation could lead to a significantly larger error in the estimated position. An alternative metric that takes this into account is the root mean square error (RMSE), which is defined as:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (y_i - x_i)^2}{n}}$$

Note that for the roar recordings, we do not directly have ground truth TDOAs available, so for this we would have to rely on manual estimation.

Position Error. The localization algorithm returns a three-dimensional position estimate from the estimated TDOAs, which again depend on the three-dimensional position of the howler monkey. Since we only have measurements of their two-dimensional positions, we simply ignore the z-coordinate estimate. A position is then defined as a point $p = (p_1, p_2)$ in the plane. For measuring the error e of the position, it is common to take the Euclidean distance between the estimated $x = (x_1, x_2)$ and ground truth $y = (y_1, y_2)$ position. This simply comes down to:

$$e = \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2}$$

If we have a set of n samples, we define the mean position error $\bar{e}$ as:

$$\bar{e} = \frac{\sum_{i=1}^{n} e_i}{n}$$

To make statements whether our methods perform significantly better than the cross-correlation methods for the localization, we need to do statistical significance testing. Suppose we have two algorithms a_1 and a_2 and n test samples. Let $e_{i,1}$ and $e_{i,2}$ denote the position errors made by a_1 and a_2 respectively on test instance i. Let $\bar{e}_1$ and $\bar{e}_2$ be the mean position error of a_1 and a_2 respectively. We can define the null (H_0) and alternative (H_a) hypothesis as follows:

- H_0: There is no difference in performance between the two algorithms (i.e., $\bar{e}_1 = \bar{e}_2$).
- H_a: There is a difference in performance between the two algorithms (i.e., $\bar{e}_1 \neq \bar{e}_2$).

Since we have a pair of errors on each test instance, we need a paired statistical test. Given our small sample size (eight roars), we can not assume that the errors are (approximately) normally distributed (e.g., $d_i = e_{i,1} - e_{i,2}$ is the difference in errors on instance i). We will therefore use the Wilcoxon signed-rank test, which tests whether the median difference between the paired observations is zero or not. For the significance level, we choose the commonly used $\alpha = 0.05$.

Geometric Dilution of Precision. Other than the quality of the TDOA estimates, the accuracy of the position estimate is mainly determined by the geometry of the source location with respect to the microphones, known as geometric dilution of precision (GDOP). It is a concept to indicate how errors in measurements (in our case TDOAs) affect the position estimate. Microphones that are poorly placed (e.g., closely spaced or collinear) typically have a high GDOP value where small errors in the measurements can cause large position errors.

While our measurements are TDOAs, the localization algorithm we use is in fact TOA-based. It essentially treats our TDOA measurements as pseudo-TOAs and then solves the position estimate with its corresponding time offset. Therefore, we consider the formulation of the GDOP that matches this mathematical structure, as also provided by Bancroft [12]. Using the same notation as earlier, the pseudorange for a microphone i is defined as:

$$\rho_i = \sqrt{(X_i - x)^2 + (Y_i - y)^2} + b$$

where b is the unknown clock offset. We compute the Jacobian matrix, which consists of the partial derivatives of these pseudoranges, by:

$$H = \begin{bmatrix} \frac{\partial \rho_1}{\partial x} & \frac{\partial \rho_1}{\partial y} & \frac{\partial \rho_1}{\partial b} \\ \frac{\partial \rho_2}{\partial x} & \frac{\partial \rho_2}{\partial y} & \frac{\partial \rho_2}{\partial b} \\ \vdots & \vdots & \vdots \\ \frac{\partial \rho_M}{\partial x} & \frac{\partial \rho_M}{\partial y} & \frac{\partial \rho_M}{\partial b} \end{bmatrix} = \begin{bmatrix} \frac{x - X_1}{R_1} & \frac{y - Y_1}{R_1} & 1 \\ \frac{x - X_2}{R_2} & \frac{y - Y_2}{R_2} & 1 \\ \vdots & \vdots & \vdots \\ \frac{x - X_M}{R_M} & \frac{y - Y_M}{R_M} & 1 \end{bmatrix}$$

We then obtain the covariance matrix by:

$$Q = (H^T H)^{-1} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xb} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{yb} \\ \sigma_{xb} & \sigma_{yb} & \sigma_b^2 \end{bmatrix}$$

where σ_x^2 and σ_y^2 are the variances in the x- and y-direction estimates respectively, σ_b^2 is the variance in the clock bias estimate, and the remaining elements are the covariances. The diagonal elements (σ_x^2, σ_y^2) basically indicate the positioning uncertainty along each axis and σ_b^2 indicates the time uncertainty. The GDOP is then defined as:

$$GDOP = \sqrt{\text{trace(Q)}} = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_b^2}$$

The interpretation of these values is listed in the table below.

GDOP	Rating	Description
< 1	Ideal	Highest possible precision. Unlikely in practice.
$1 - 2$	Excellent	Very good accuracy.
$2 - 5$	Good	Acceptable for most applications.
$5 - 10$	Moderate	Noticeable degradation in accuracy.
$10 - 20$	Poor	Very low precision. Positions should only be used as rough indication.
> 20	Very poor	Positions should be discarded.

Chapter 5

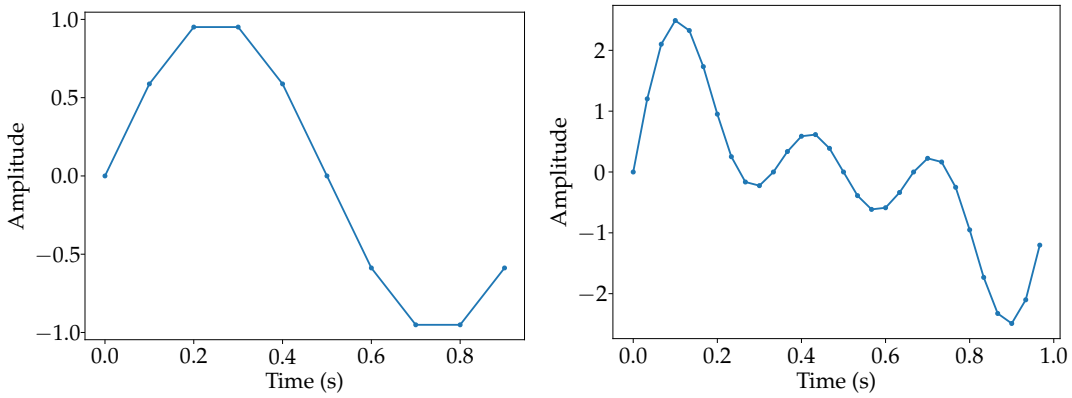
Experimental Results

In this chapter, we present, interpret and discuss the experimental results of the matching algorithms described in Chapter 3 applied to signal data. It is divided into three broad sections:

- Matching of simple signals representing x-monotone curves in space.
- The TDOA estimation of roars.
- The localization of roars using these TDOA estimates.

Throughout the experiments, we use labels to refer to the methods being examined, which can be found in the table below.

Label	Method
HD_PNT	Hausdorff distance for point sets L_1 and L_2 (in case of equal performance)
HD_L**X**	Hausdorff distance for point sets ($L_{\mathbf{X}}$)
HD_SEG	Hausdorff distance for line segments (L_2)
CON_FD	Continuous Fréchet distance (L_2)
DIS_FD	Discrete Fréchet distance (L_2)
EMD	Earth Mover's Distance (L_1 unless otherwise specified)
CR_CRR	Cross-correlation
KM_**X**	Weighted K-means with aggregations $\mathbf{X} \in \{\text{MAX}, \text{SUM}, \text{AVG}\}$
GI	Greedy Insertion



(A) Single frequency 1 Hz

(B) Sum of frequencies 1,2,3 Hz

FIGURE 5.1: Shapes of sine waves.

5.1 Matching Simple Signals

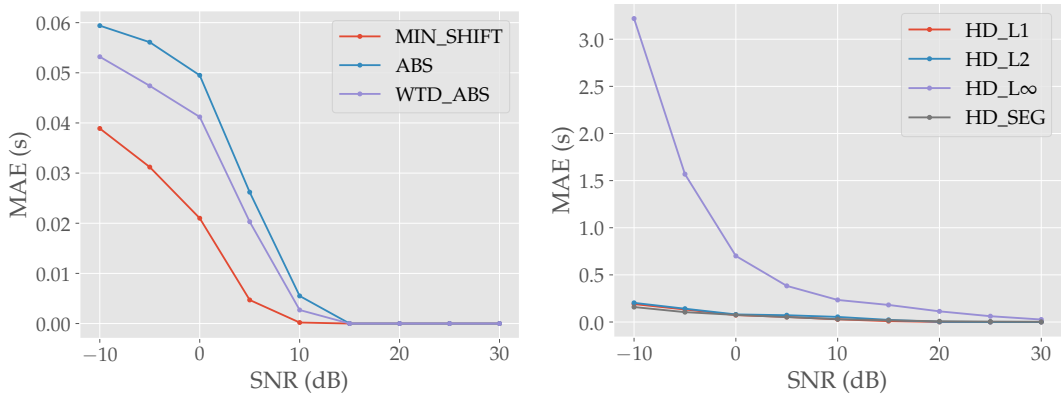
Since the partial matching of signals is achieved by matching a template segment shape with the partial shapes of the target segment, the matching of such a partial shape directly influences the quality of the TDOA estimation. Therefore, we consider simple signals with distinct shapes and observe how distortions in the amplitude values of the target signal affect the optimal alignment with the unaffected template signal. This could give an indication of the robustness for each method and how they compare. Note that we do not consider the spectrogram representation since our choice of signal representation are mainly x-monotone waveform curves. Besides, a spectrogram is more appropriate for complex signals with changing properties of frequencies and amplitudes over time, which applies more to real-world signals.

5.1.1 Setup

The simplest and most fundamental sound signal is a sine wave, characterized by its smooth and harmonic shape. We start with a sine wave that has an amplitude of $A = 1$, a frequency of $f = 1$ Hz, a phase of $\phi = 0$, and a duration of 1 second, which corresponds to a single cycle. By the Nyquist-Shannon sampling theorem, we need a sampling rate of at least 2 Hz to capture the information of this signal, but this does not provide us a visually smooth waveform. For this reason, we choose a sampling rate of 10 Hz, resulting in a total of 10 samples. This also appears to be the minimum amount of samples required to obtain the recognizable shape (see Figure 5.1a). Since the noise added onto the target signal is randomly sampled from a Gaussian distribution, we have to average the results over multiple instances. At each noise level, we therefore take the average over 1000 instances and each algorithm is tested on the same set of instances for proper comparisons. As our main goal is to observe the behavior of the algorithms on small distortions of the signal, we ignore the attenuation effect to keep it simple and set $\alpha = 1$. For the continuous and discrete Fréchet distance, we set $\varepsilon = 0.00001$ as threshold for the binary search, which allows for a nearly exact solution.

5.1.2 Preliminary Experiments

EMD Weights. One other practical consideration is how we deal with the weights in the EMD algorithm. Since the loud parts of the signal are considered more important,



(A) EMD weight strategies

(B) Hausdorff distance variations

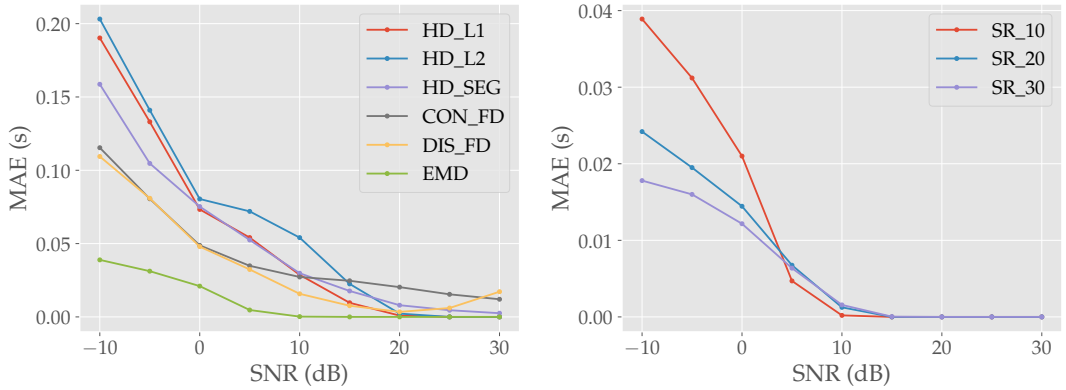
FIGURE 5.2: Preliminary experiments on matching performance ($f = 1$ Hz).

we can simply model the amplitude values of the data as the weights of the points. We expect that the total weights of the two points are (in general) unequal and will therefore be partially matched. However, there is one problem with this approach as the amplitudes can take negative values, whereas the weights must be non-negative. There are three potential solutions for this:

- **MIN_SHIFT:** Shift all the amplitude values by the minimum amplitude value (of both signals).
- **ABS:** Take the absolute values of the amplitudes.
- **WTD_ABS:** Take absolute values, but positive amplitudes get higher weights (i.e., by doubling them).

The latter option allows to still differentiate between positive and negative amplitudes. The results for each strategy are shown in Figure 5.2a. It shows that for a SNR of ≥ 15 dB, each strategy performs equally well, but when the noise further increases, the strategy of shifting the amplitude values outperforms the other strategy on average. This is in line with expectations as this best preserves the original shape by keeping the relative amplitude differences consistent. It is also worth noting that the weighted absolute values in this situation performs better than the regular approach, confirming that the differentiation between the positive and negative amplitudes has slightly improved.

Hausdorff Variations. We also performed a preliminary experiment for a first comparison between each Hausdorff distance variation, presented in Figure 5.2b. This shows that, on average, the Chebyshev metric has a significantly larger matching error compared to any of the other metrics at each noise level. Recall that the Chebyshev distance between two points is the maximum distance along any coordinate dimension, so the distances from the other coordinate dimensions are basically ignored in the final measure, which is not the case in the other variations. As the noise further increases, the distances of those other coordinate dimensions also appear to become increasingly important for a proper matching. This makes this metric less suitable for our particular matching application where precision in time differences is important. For this reason, we omit this variation in the remaining experiments.



(A) Sample rate of 10

(B) Effect of sample rate on EMD

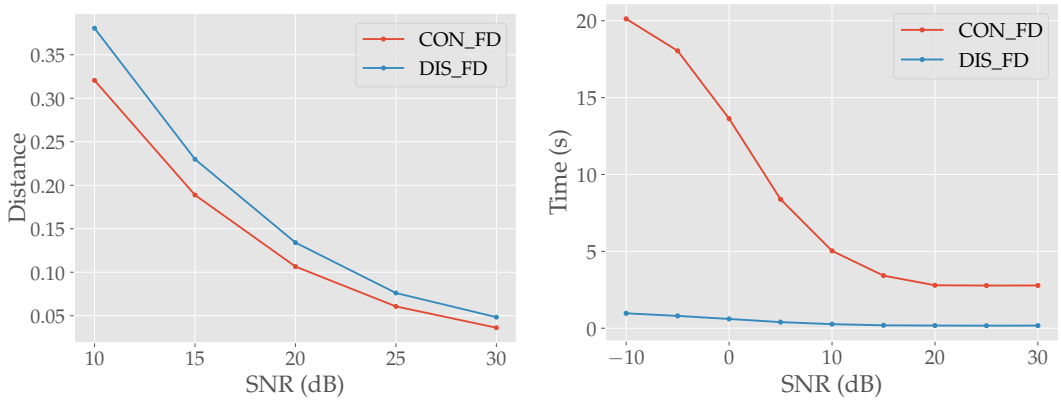
FIGURE 5.3: Matching performance on a sine wave ($f = 1$ Hz).

5.1.3 Matching Performance

We now move our attention to the matching results of all the algorithms shown in Figure 5.3a.

Hausdorff Distance. It stands out that the Hausdorff distance for the L_1 metric performs on average consistently better compared to the L_2 metric variant at the higher noise levels. Given also our earlier observation of the L_∞ metric performing worse than the L_2 metric, this might indicate that higher-order L_p-norms as underlying distance are more vulnerable to distortions in the signal, which leads to larger matching errors. Another interesting observation is that the Hausdorff distance for segments seems to perform, on average, noticeably better at the larger noise levels (< 0 dB), equally well on the middle noise levels ($0 - 10$ dB), and worse at the lower noise levels (> 10 dB), compared to the Hausdorff distance for L_1. It shows the benefit of the added precision through interpolation of consecutive points in settings with larger noise levels. This is even more confirmed if we take a glance at the performance of the continuous Fréchet distance, which shows further improvement as it adds another precision layer by taking the course of the curves into account as well. However, in settings with lower noise levels, the noise has a larger negative impact, which we would expect with smaller distortions and higher precision.

Fréchet Distance. Continuing on the topic of the Fréchet distances, the continuous variant generally does not improve on the discrete variant and, at best, matches the performance at the larger noise levels (< 10 dB). Figure 5.4 shows the solution quality and wall-clock time comparisons between them. Since the discrete Fréchet distance is restricted to only the vertices of the curves and is in that regard an approximation of the continuous variant, it is expected that the continuous distance finds solutions with slightly lower distances, which is in accordance with our observations. Although there is some improvement in the solution in terms of its distance, it surprisingly does not lead to better matchings at the larger noise levels on average, indicating that the vertices alone are typically sufficient enough and that we possibly reached a limit on the performance gain for matchings that optimize a maximum distance in space. In terms of computation time, we already showed that the algorithm for solving the decision problem of the continuous variant is a factor $O(m + n)$ slower in time complexity compared to the discrete variant. It shows that this, in combination



(A) Estimated distances (B) Wall-clock times

FIGURE 5.4: Comparisons between continuous and discrete Fréchet distance ($f = 1$ Hz).

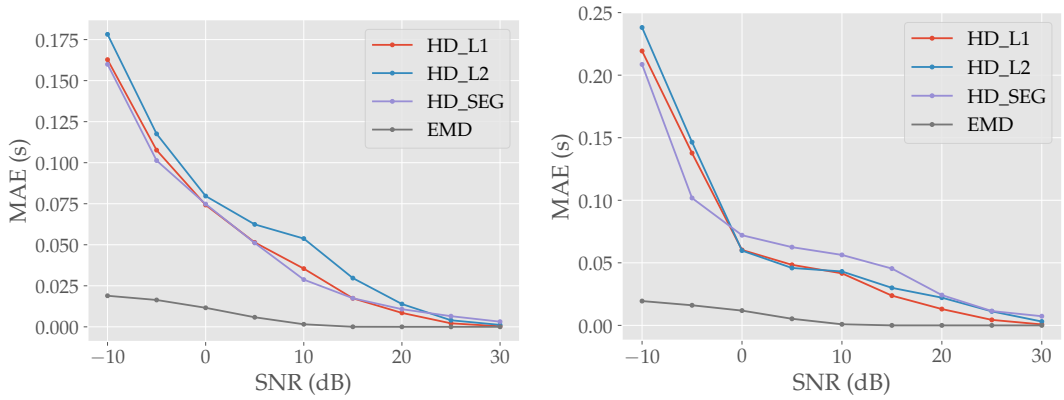
Method	SR_10	SR_30	*Increase (%)*
HD_L1	0.00272	0.0237	771
HD_L2	0.00749	0.0840	1021
HD_SEG	0.0769	1.029	1238
EMD	0.00134	0.00374	179

TABLE 5.1: Rough indication of wall-clock times (s) on a single matching for two different sample rates. Averages over all 9000 instances of each noise level.

with the more expensive intersection operations that need to be done, has a large impact on the wall-clock times, especially with higher levels of noise. The reason for this lies in the fact that higher noise levels lead to curves with higher distances. At the same time, we have set a fixed lower bound of zero, which consequently results in more iterations being needed for the binary search to converge. For the continuous Fréchet distance, this significantly increases the computation time by multiple seconds, whereas the impact on the discrete Fréchet distance stays relatively low. Since we already observe relatively high computation times for a single matching between two curves of 10 points and no remarkable improvement compared to its discrete counterpart, we omit the continuous variant in any further experiments.

For reference, the wall-clock times of the other methods are shown in Table 5.1, which in contrast to the Fréchet distance, do not depend on the specific SNR value. For the Hausdorff distance, the more complex it gets, the worse it scales. The EMD scales relatively well in comparison.

EMD. One final interesting observation is that the EMD outperforms all other algorithms, especially at the larger noise levels. We must take into account the fact that the optimal matching in this case always occurs when the coordinates of the two points align. This means that there is either a perfect matching with time error of 0 seconds or an imperfect matching with a time error of at least $1/10 = 0.1$ seconds. Since we observe average errors of lower than 0.05 seconds on all noise levels, this means that it succeeds more than 50% of the time in correctly aligning the curves at any noise level and (nearly) all of the time for smaller noise levels (≥ 10 dB). Given this observation of the sample rate playing a role in the matching quality for the



(A) Single frequency 1 Hz

(B) Sum of frequencies 1,2,3 Hz

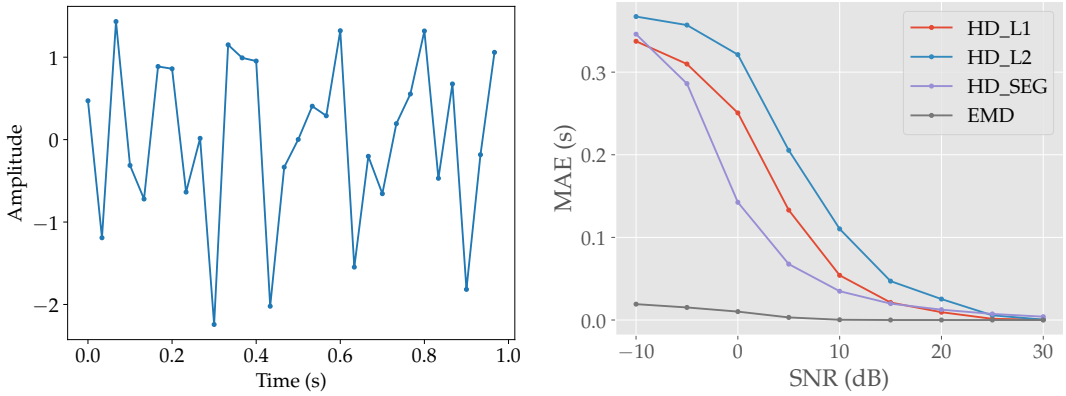
FIGURE 5.5: Matching performance on sine waves with a sample rate of 30.

EMD, it might be interesting to investigate the effect of larger sample rates. This is shown in Figure 5.3b. It becomes immediately clear that a larger sample rate can be beneficial in settings with higher noise levels (≤ 0 dB), but it seems to sacrifice a little in the middle noise levels ($5 - 10$ dB) again. More samples increases the likelihood for a wrong matching in settings where ambiguity starts to increase, which typically happens at the middle noise levels. For the lower noise levels (≥ 15 dB), there is no difference as it always finds the correct alignment for any sample rate.

5.1.4 Effect of Signal Type

All the previous observations were based on a simple sine wave, but these might not always hold for other signal types. Therefore, we also consider other signals that have distinct structures. Due to computational restrictions, we ignore the Fréchet distance methods in this section.

Sum of Sine Waves. All sound signals can essentially be represented as the sum of one or more sine waves, each with their own set of parameters. If we increase the frequency, we obtain more cycles within the same duration. Summing the sine waves of frequencies 1, 2 and 3 Hz results in a more interesting shape shown in Figure 5.5b. Note that this requires a sample rate of 30 Hz since our maximum frequency has become 3 Hz. To ensure fair comparison with our simple sine wave as well, we evaluate the performance on a sine wave of 30 samples, of which the results are given in Figure 5.5a. We observe fairly similar results to the sine wave of 10 samples, but one noticeable difference is that the Hausdorff distance for point sets starts to align more with the segments variant. More samples better approximates the original curve and therefore the impact of the added precision through interpolation naturally degrades. The results of our complex signal are shown in Figure 5.5b. One interesting observation is that the L_2 metric matches the L_1 metric at the middle noise levels ($0 - 10$ dB), whereas we clearly saw a different pattern with the simple sine wave thus far. Another remarkable change concerns the segments variant, which again seems to perform better at the larger noise levels (< 0 dB), but now performs noticeably worse at both the middle and lower noise levels (≥ 0 dB). It seems that for increasingly complex structures, the added precision of the segments variant becomes



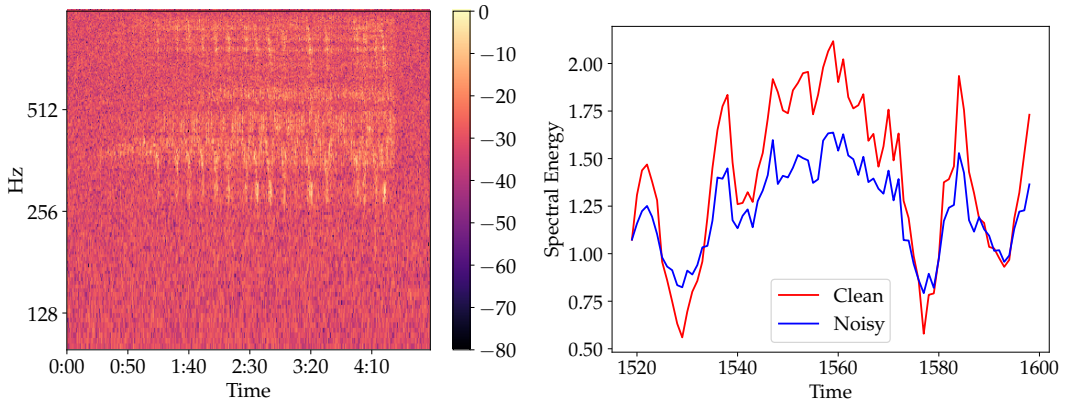
(A) Shape of the signal

(B) Matching performance

FIGURE 5.6: White noise signal with a sample rate of 30.

more vulnerable to small distortions, but when the noise dominates the signal, it remains a beneficial feature.

White Noise. Another distinct signal type is a white noise signal, which in contrast to sine waves, lacks any structure. The signal is sampled from a Gaussian distribution with zero mean and standard deviation of one, similarly to how noise is modeled. An example of this is shown in Figure 5.6a, which also matches the sample rate of our previous signal types. We can use this specific instance and compare the matching performance again. The results of this are shown in Figure 5.6b. We observe a convergence behavior, where increasing noise levels no longer significantly increase the matching error. The gap between the Hausdorff distance for the L_1 and L_2 metrics has become more apparent again. Another noticeable difference is the segments variant that significantly improves on the L_1 metric now in the middle noise levels ($0 - 10$ dB). This could mean that the added precision of the Hausdorff distance for segments becomes more important with signals that have less structure, which seems to be in line with our earlier observation for a more complex structured signal where it becomes less important. A similar statement could be made for the L_1 and L_2 metrics, where the L_1 metric seems to perform significantly better for signals that have less structure, but the L_2 metric starts to match its performance with more complex structures at the noticeable noise levels.



(A) Spectrogram (B) Energy envelopes at largest peak (normalized)

FIGURE 5.7: Example of roar with artificial noise (SNR = 5dB and $\alpha = 0.8$; spectrogram of clean version in Figure 4.3).

5.2 Roar TDOA Estimation

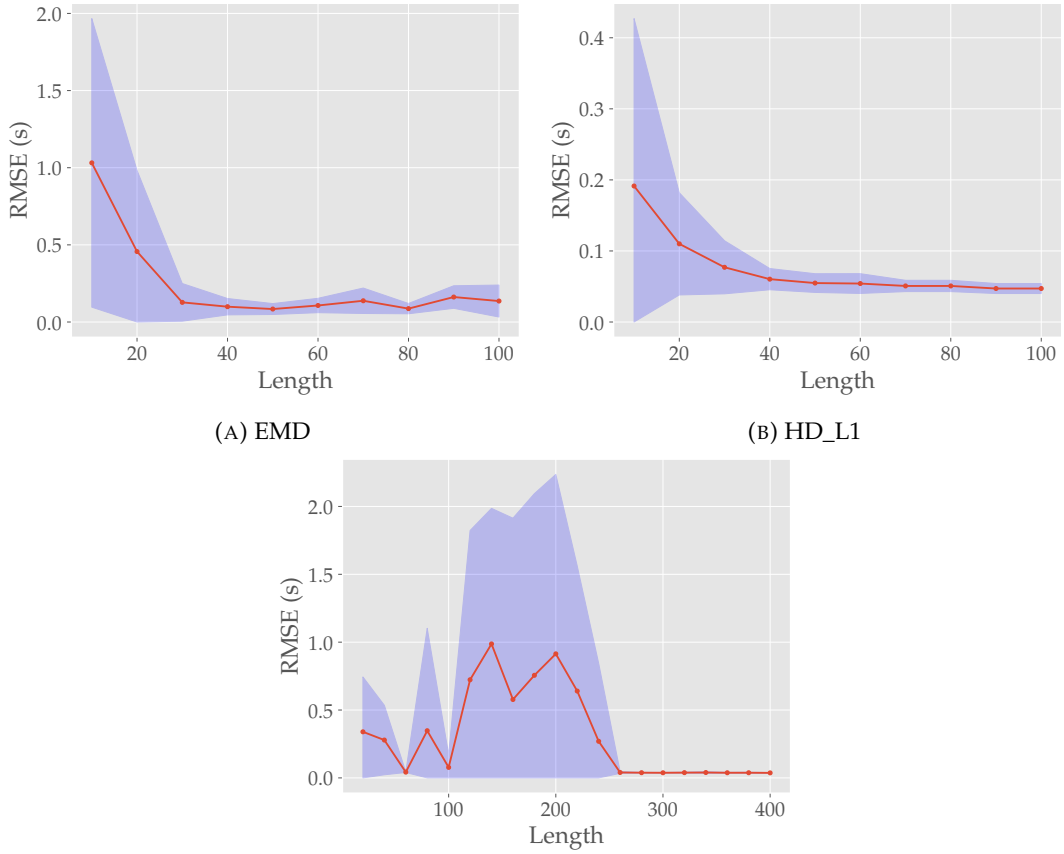
In this section, we evaluate the performance of our methods on the TDOA estimation if the signals represent roars captured from recorders in the field, and compare them to the cross-correlation method. There are multiple factors that could affect the quality of the TDOA estimation, including the template length, the noise robustness characteristics of the respective method, and the input representation of the signal. We investigate each of those aspects to empirically find an optimal setting for this, which should also provide the best possible performance in the localization.

General Setup. Although we try to incorporate all the algorithms described in this thesis, we can only focus on a small subset of them due to poor scalability. Our first candidate is the EMD, given the fact that it showed the best performance for a single matching in noisy settings, so we expect that if we can improve this algorithm in a certain experimental setting, that this consequently has a positive effect on the other algorithms as well. However, we should be cautious due to the different nature of the Hausdorff and Fréchet distances. Another candidate is therefore the Hausdorff distance for point sets with underlying metric L_1 due to its better performance (especially for structured shapes and reasonable noise levels) compared to the other variants. To ensure fair comparison, the cross-correlation is also applied to the envelope representation (unless otherwise specified).

5.2.1 Effect of Template Length

The template length is a key parameter in the partial matching process. A larger template allows for a more precise matching in most cases, but can have a significant impact on the runtime. Therefore, the challenge is to find an optimal balance between a sufficient template length that allows for a proper comparison with the target segment and such that the partial matching can be computed in a reasonable amount of time.

Setup. We take the recording of a roar with the highest resolution as both our template and target signal and apply noise and shifts to the target copy. Using our noise model, we set a SNR of 5 dB and an attenuation of 0.8. This results in an energy



(A) EMD

(B) HD_L1

(C) CR_CRR

FIGURE 5.8: Effect of the template length on the matching performance (SNR = 5dB and $\alpha = 0.8$). The red line shows the mean and the blue area shows the 95% confidence interval based on the RMSE of five peaks. The RMSE of a single peak is determined over 100 instances.

envelope that is difficult enough to match with the original shape, while still being comparable (see Figure 5.7). Note that the spectrogram of the noisy signal gets larger energies in areas of lower energies in the original signal. Our noise model adds equal power at all frequencies, which reduces the contrast of the original signal. This naturally affects the energy envelope as well.

Results. The results are shown in Figure 5.8. We observe that for both the EMD and Hausdorff distance, the error gradually decreases as the template length increases until a point of convergence where no significant improvements are made. This happens at a template length of around 40 (which is the same as our chosen parameter value). In fact, a larger template seems to increase the error again for the EMD. The cross-correlation method requires a considerably larger template length with convergence occurring at 260. This may be explained by the geometric nature of the EMD and Hausdorff distance as they directly take into account the spatial relationships of the points, whereas the cross-correlation ignores this and simply creates a summary value based on a dot product of amplitude values. This is computationally more efficient, but requires a larger template length in return for a reliable comparison.

Effect of Noise Setting. We also experimented with a larger noise level setting (SNR = 0dB and $\alpha = 0.6$) and a lower noise level setting (SNR = 10 dB and $\alpha = 1$) to observe how this affects the optimal template length compared to our previous (medium) noise setting. The results can be found in Figure B.1 and B.2 of Appendix B respectively. For the high noise level setting, we still observe a similar convergence pattern for the EMD and cross-correlation at the same lengths as with our medium setting, although the errors are unsurprisingly on a higher scale and more fluctuating. For the low noise level setting, the EMD seems to converge slightly earlier at 20, whereas the other algorithms follow the same pattern as our medium setting, with lower scale errors and more certainty. This confirms that in other noise settings, it is not necessary to adapt the template length. Except possibly for the cross-correlation, which showed a small jump in error again at 320 with the larger noise setting.

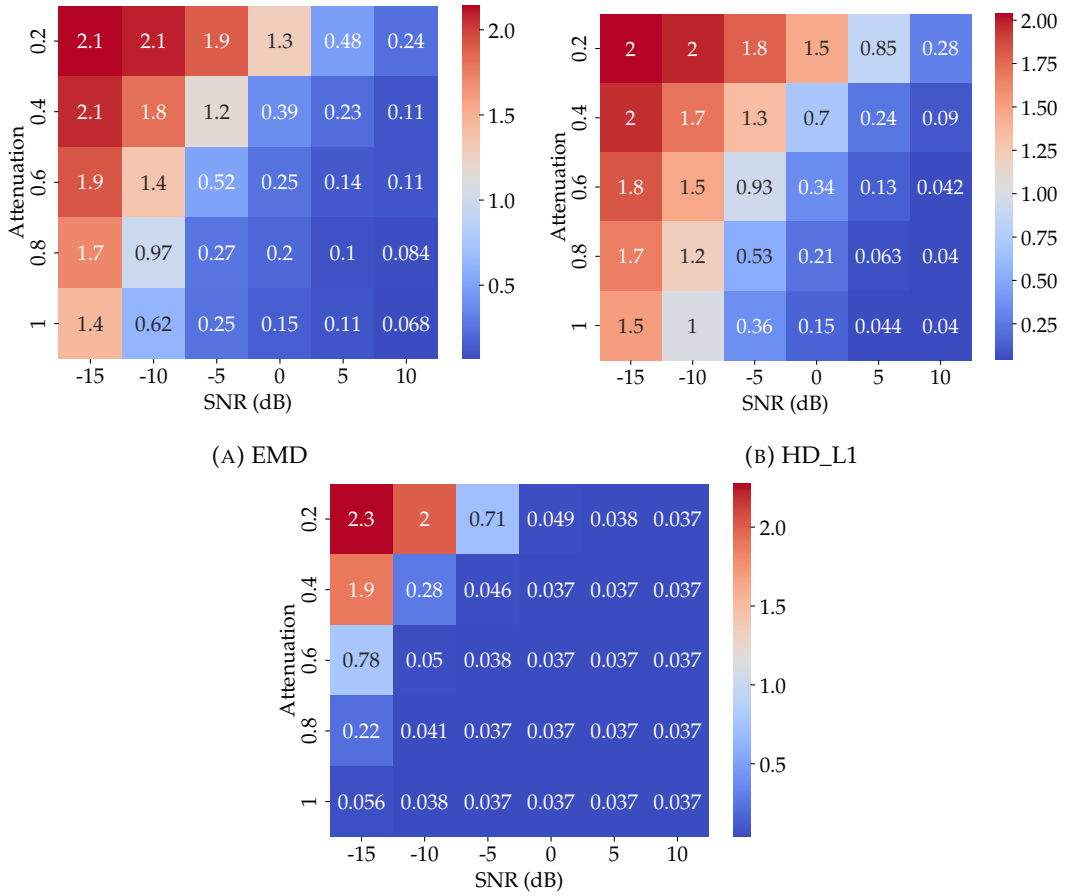
5.2.2 Noise Robustness Analysis

A method that is more robust to noise should in theory provide more accurate TDOAs. We have already investigated the performance of a single matching for simple signals in noisy settings, however, the situation with roars is different in various aspects. The spectral energy envelopes of roars clearly have their own distinct patterns which are vastly different from clean sine waves or white noise signals. Moreover, the TDOA estimation is achieved by a partial matching, since we have no foreknowledge where the template is exactly (or approximately) represented within the target segment. This can drastically increase the TDOA errors due to a wrong matching with a partial shape that is located at a distant timestamp.

Setup. Similarly as in the previous section, we take the recording of a roar with the highest resolution as both our template and target signal and apply noise and shifts to the target copy for a variety of noise and attenuation levels.

Results. The results are shown in Figure 5.9. First of all, we observe expected behavior where degradation of the signal quality in terms of both the SNR and attenuation results in non-decreasing (and generally increasing) TDOA errors. The Hausdorff distance seems to perform slightly better in the regions representing the lowest noise levels (attenuation $0.6 - 1$ and SNR $5 - 10$ dB), but moving outside this range, the EMD seems to become more robust. When the noise starts to dominate the signal (SNR ≤ 0 dB), the performance gap narrows to a point where it becomes equally difficult for them. On the other hand, the cross-correlation seems to be significantly more robust overall to the noise when we consider the entire signal as template, which is still computationally more feasible than most of our methods at their current template lengths. We should keep in mind our earlier observation that a larger template length does not necessarily improve the TDOA estimation at different noise settings, so while it seems like an unfair comparison at first sight, this is likely the best that can be achieved for each individual method.

Effect of Template Length. To experimentally demonstrate this, we performed the same evaluation on other template lengths as well, presented in Figure B.3 of Appendix B. It shows that if we match the template length of cross-correlation with the other methods (40), then it noticeably struggles at the lower noise levels (0.27 vs 0.037) as expected. If we set the template length at our earlier observed convergence point for the cross-correlation (around 300), then the cross-correlation performs slightly

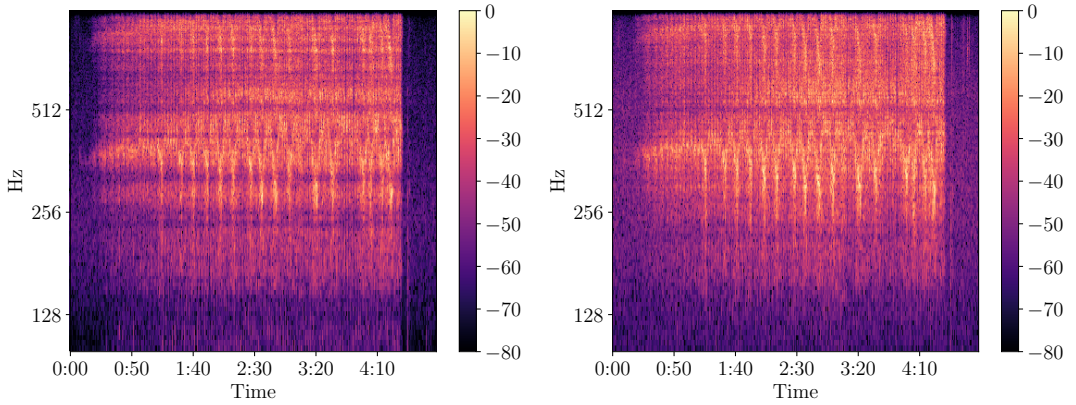


(A) EMD

(B) HD_L1

(C) CR_CRR

FIGURE 5.9: Noise robustness on the roar with the highest resolution. Average RMSE values over the five largest peaks. The RMSE of a single peak is determined over 100 instances.



(A) Highest resolution

(B) Second-highest resolution

FIGURE 5.10: Same roar captured by two different microphones (LP3-D (I)).

worse at the higher noise levels than when we take the entire signal, confirming that a higher template length is always more beneficial for this method. The EMD however shows significantly worse performance overall in this case, which confirms our earlier observation that a larger template length increases the error again. If we instead set the template length of all methods to 100 and consider only the largest peak for a subset of the noise levels (due to computational restrictions), we again observe the cross-correlation outperforming the other methods (see Figure B.4 of Appendix B).

5.2.3 Real Estimation

Now that we have analyzed the noise robustness of some of our methods, the next step is to evaluate the performance on roars captured in the field by actual different microphones. While our noise model tries to accurately capture the way noise affects the signal, it is clear that this still deviates from real-world observations. The analysis in the previous section provides an indication of their robustness, but we still expect slightly different results in practice. Moreover, it assumes that the original template signal is a clean roar, which is typically not the case even for the recording of a roar with the highest resolution. A final limitation is that, while the roars share similar acoustic characteristics, there are still differences among them and each method might respond differently to those.

Setup. Since the recordings of the roars all have varying quality that we can not directly describe in terms of the SNR or attenuation parameters, we focus on the performance in a best-case scenario setting. This is done by taking a roar that has been captured best by at least two recorders, as measured by the RMS, where we again take the one with the highest resolution as reference. The roar satisfying this condition comes from the LP3-D setup (see Figure 5.10). We expect that this setting should be relatively easy for our methods as a more difficult setting is not expected to improve the performance either. By manual inspection, the real TDOA appears to be -0.799 seconds, which is also the output of the cross-correlation method when applied to the waveform representation. When it is applied to the envelope representation, the output is -0.768 seconds (due to its lower time resolution). We will look at a top-5 of largest peaks. When estimating the TDOA, we often take the matching with the lowest distance and return its corresponding translation as output TDOA.

Peak	HD_PNT		HD_SEG		DIS_FD ($\varepsilon = 0.01$)		EMD	
	t	dist	t	dist	t	dist	t	dist
1	$\mathbf{-0.768}$	0.472	-0.956	0.292	-0.749	0.495	-0.896	**0.120**
2	$\mathbf{-0.768}$	0.380	-0.736	0.221	-0.883	0.552	-0.64	0.173
3	-0.768	0.288	$\mathbf{-0.823}$	0.168	-0.749	0.324	-0.896	0.348
4	$\mathbf{-0.768}$	0.426	-0.703	0.280	-0.750	0.448	-1.28	0.375
5	-0.768	**0.266**	$\mathbf{-0.799}$	**0.119**	-0.744	**0.324**	-0.64	0.156

TABLE 5.2: Top-5 largest peaks matchings for a single roar (LP3-D (I)) between two high-resolution recordings. For each peak, the best estimation compared to true TDOA (-0.799) is in bold. For each algorithm, the smallest distance is in bold.

Results. For the discrete Fréchet distance, we need to determine a suitable ε value. We experimented with various values at the largest peak (see Table B.1 of Appendix B) and found $\varepsilon = 0.01$ to have the best quality-runtime ratio.

The results of the matching are shown in Table 5.2. One interesting observation is that the Hausdorff distance for the L_1 and L_2 metrics show equivalent matching performance. For both metrics, the optimal matching between the template and the corresponding partial target shape is (in most cases) when they perfectly align in time and where the maximum distance is defined between two points at the same x-coordinate. This means that the distance simply becomes the absolute distance of the other coordinate dimension, which is why they have equal distances as well. One explanation for this are the unit differences between the coordinate dimensions. The contribution of the other component difference may be (significantly) less, so by aligning the two shapes, there is always another point on the same x-coordinate and the distance is then only defined by the smaller other component difference.

Another interesting observation between them is that they not only consistently return the same translation for all five peaks, but also frequently obtains the best translation estimates of all the methods. The best estimation, however, was found by the segments variant, which would have also been chosen as output TDOA because of its smallest matching distance, but seems to be less consistent over the other four peaks in comparison. The segments variant therefore seems to have the potential to find more accurate TDOAs, but this largely depends on the chosen template segment, which is not necessarily the one corresponding to the largest peak. The discrete Fréchet distance shows fairly consistent results, but the estimation quality does not exceed that of the Hausdorff distance for points while still requiring significantly more runtime (1 hour vs 7 seconds, on average). For the EMD, we observe the most variability in the estimations, but despite this, still chooses its closest estimation out of those five.

Based on these observations, we would expect the Hausdorff distance for point sets to perform best in TDOA estimation overall, with an occasional better estimation from the segments variant. The EMD is expected to perform the worst due to both its inconsistency and larger errors. Although we should keep in mind that the noise level is supposedly quite low in this example and we already observed that the EMD can perform better with higher noise levels. However, if we assume that the noise is relatively constant across the recordings, then the recordings with the highest resolutions often correspond to the ones with the higher SNR and lower attenuation (we also observe this by inspection of the spectrograms in Figure B.5 of Appendix B).

Peak	t	dist	template_error	target_error	run_time
1	−0.768	1.563	6.046	5.315	6h42m
2	−0.899	2.431	6.811	8.959	6h24m
3	−1.175	1.995	4.869	4.689	6h9m
4	−0.512	1.019	7.798	5.165	6h8m
5	−1.152	2.157	5.435	5.546	6h16m

TABLE 5.3: Hausdorff distance for triangles using a greedy insertion approximation at 80 vertices (baseline is −0.768 and true is −0.799).

So by always taking these when estimating TDOAs, our expectations become more likely.

Another Roar. To additionally support our statements, we performed the same evaluation on a different high-quality roar, which was also from the LP3-D setup. The results can be found in Table B.2 of Appendix B, which shows similar results. For some peaks, the discrete Fréchet distance now returns the closest estimate instead of the Hausdorff distance for point sets, but by a marginal difference (less than 0.02 seconds). This still does not outweigh the additional computational cost. If we also look at the performance on a low-resolution recording (of which the roar has still been sufficiently captured) in Table B.3 of Appendix B, the EMD returns closest estimates in some peaks as expected. The Hausdorff distance for segments remains to find the closest estimates in both cases as well.

5.2.4 Spectrogram Approximations

We now consider spectrogram approximations instead of our envelopes, which might be able to improve our previous obtained results.

Setup. We use the same setup as in the previous section and evaluate the performance on our methods supporting this three-dimensional input representation. We also define a baseline, which is simply the best translation estimate from the envelope representation of its corresponding method (i.e., −0.768 for the Hausdorff distance and −0.896 for the EMD).

Triangles. The greedy insertion method aims to accurately approximate the original terrain using as few triangles as possible. This makes our Hausdorff distance method for triangles an ideal candidate to start with. Often times, the termination condition for the approximation is based on an error measure, but given the relatively large runtime of our method, we should set this based on the number of vertices instead. We experimented with various number of vertices at the largest peak (see Table B.4 of Appendix B) and 80 vertices appeared to match our baseline, which already takes over six hours to run, so we try to keep it at a minimum.

The results for all five peaks are shown in Table 5.3. Contrary to the point sets method on the envelope representation, we observe inconsistent results where the best translation estimate is obtained at the largest peak, but the translation corresponding to the smallest distance is −0.512, which deviates 0.256 seconds more as opposed to the baseline. Part of this can be explained by the approximation errors of the template

Peak	t	dist	template_error	target_error	run_time
1	-0.64	1.899	4.010	3.820	1h10m
2	-0.896	2.476	4.000	3.748	1h12m
3	-0.512	1.672	3.362	2.851	1h6m
4	-1.184	1.000	3.779	3.731	1h8m
5	-0.64	1.378	3.739	3.121	1h7m

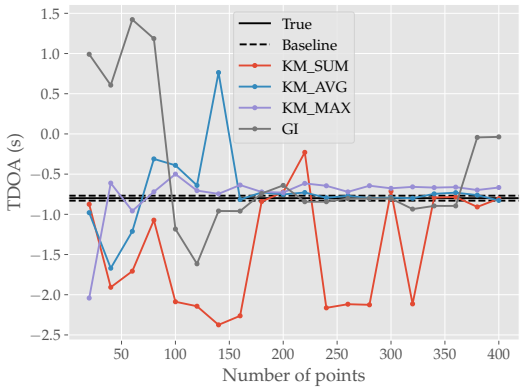
TABLE 5.4: Directed Hausdorff distance for triangles using a greedy insertion approximation (baseline is -0.768 and true is -0.799). Template consists of 200 vertices and target consists of 400 vertices.

and partial shape of the target, which represents the maximum vertical error of any point in the grid. Note that the amplitude values have no fixed range since we use RMS normalization to scale them, which complicates the interpretation of these error values. For reference, in the original terrains we observe amplitude values of around 1 ± 1.2 (Mean $\pm$ SD) and maximum values in the range $[10, 25]$ (see Table B.5 of Appendix B for a more detailed description). We observe an average error of around 6, which is still relatively high considering this range. If we allow more vertices in the approximation, we expect to observe increasingly better estimates, but this does not seem to outweigh the computational cost.

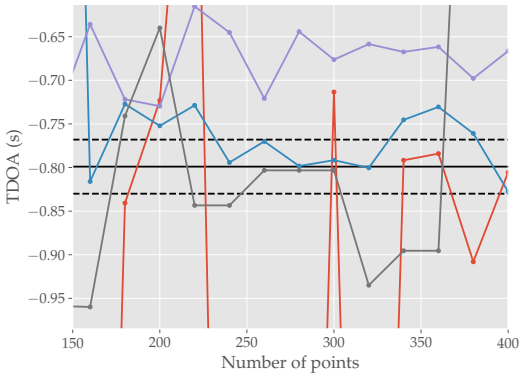
Triangles (weaker definition). One solution for this would be to consider a weaker definition of the Hausdorff distance instead. Currently, we aim to find an optimal matching in terms of the undirected Hausdorff distance between the template and each partial shape of the target segment, but we could also try to find an optimal matching from the template to the entire target by means of the directed Hausdorff distance. We increase the number of vertices from 80 to 200. Since the target segment is twice as long as our template segment, we also need to double the amount of vertices in the target approximation to compensate for this.

The results are shown in Table 5.4. We still observe inconsistent results, but the translation estimates seem to be overall in the right range as they stay fairly close to the ground truth, which is a pattern we also observe to a lesser degree in the undirected distance variant. The estimate corresponding to the lowest distance is -1.184, which clearly has the largest deviation of all of the estimates. On a positive note, we observe smaller approximation errors with values typically below 4. The template and target errors also appear to be approximately equal as well, indicating that their approximation quality is similar and matching them should be fair. While the directed Hausdorff distance is capable of matching the terrain approximations consisting of more vertices in considerably lower runtimes (1 hours vs 6 hours, on average), it thus does not significantly improve the estimates. The weaker distance definition might therefore have a larger negative impact than the gain in approximation quality.

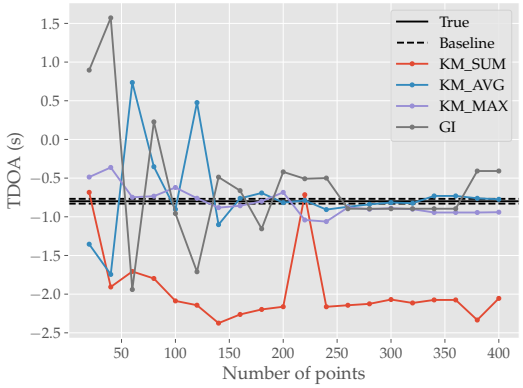
Vertices. Another solution is to keep our stronger distance definition of the undirected Hausdorff distance, but only consider the vertices of the approximations. This allows us to further increase the number of vertices as it improves the computational feasibility again. The methods supporting this representation are the Hausdorff distance for both the L_1 and L_2 metrics. For the EMD, we restrict ourselves to the L_2 metric due to limitations of the implementation being used.



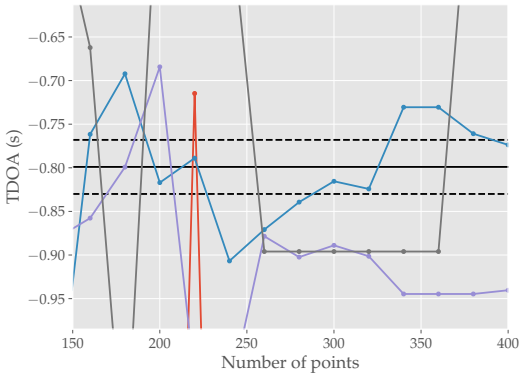
(A) HD_L1



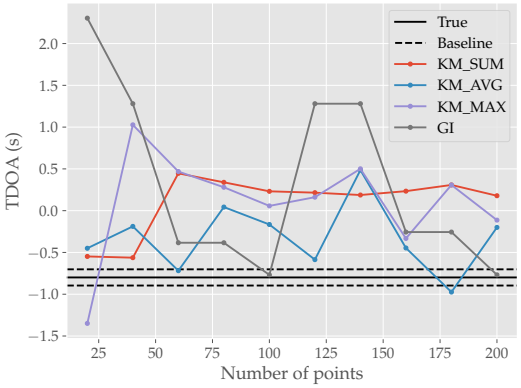
(B) HD_L1 (zoomed in)



(C) HD_L2



(D) HD_L2 (zoomed in)



(E) EMD (L_2)

FIGURE 5.11: Estimated TDOA on spectrogram approximations between two high-resolution recordings of a roar at the largest peak for varying number of points.

The results at the largest peak are shown in Figure 5.11. Contrary to the envelope representation, the EMD scales worse than the Hausdorff distance, so we had to terminate at 200 vertices while the Hausdorff distance was able to continue up to 400 vertices. We would expect to find a convergence point an increase in the number of vertices no longer improves the estimation, but for the most part, we observe unstable estimations. The K-means sum variant falls largely behind as it does not come close to the true TDOA in general, whereas the other approximations at least tend to get (and stay) close starting from a certain number of vertices (around 150) for the Hausdorff distances specifically. This happens relatively early for the K-means max variant, which summarizes their clusters by taking the most pronounced point. These are expected to be more important in the spectrogram, which consequently allows for an early sufficient matching. In contrast, the K-means average and greedy insertion require more vertices for a more accurate approximation and matching. If we look at the other peaks for the Hausdorff distance with underlying metric L_1 (see Figure B.7 of Appendix B), then we observe the exact same instability property.

Vertices (weighted L_1). Thus far, the distance measures assigned equal weight to the distances of each coordinate dimension between two vertices. If we have two vertices $v_1 = (t_1, f_1, a_1)$ and $v_2 = (t_2, f_2, a_2)$ where t, f, a represent the time, frequency and amplitude respectively, the L_1 metric for example returns a distance of:

$$|t_1 - t_2| + |f_1 - f_2| + |a_1 - a_2|$$

However, we have already seen that the unit differences affect the matching in the envelope representation, so this likely also affects the spectrogram approximations which could explain the instability in the estimates. Given this observation, the distance of a certain component should perhaps have a larger impact on the overall distance. We can turn our L_1 measure into a weighted variant that takes this into account. For weights α, β, γ, the distance is then measured by:

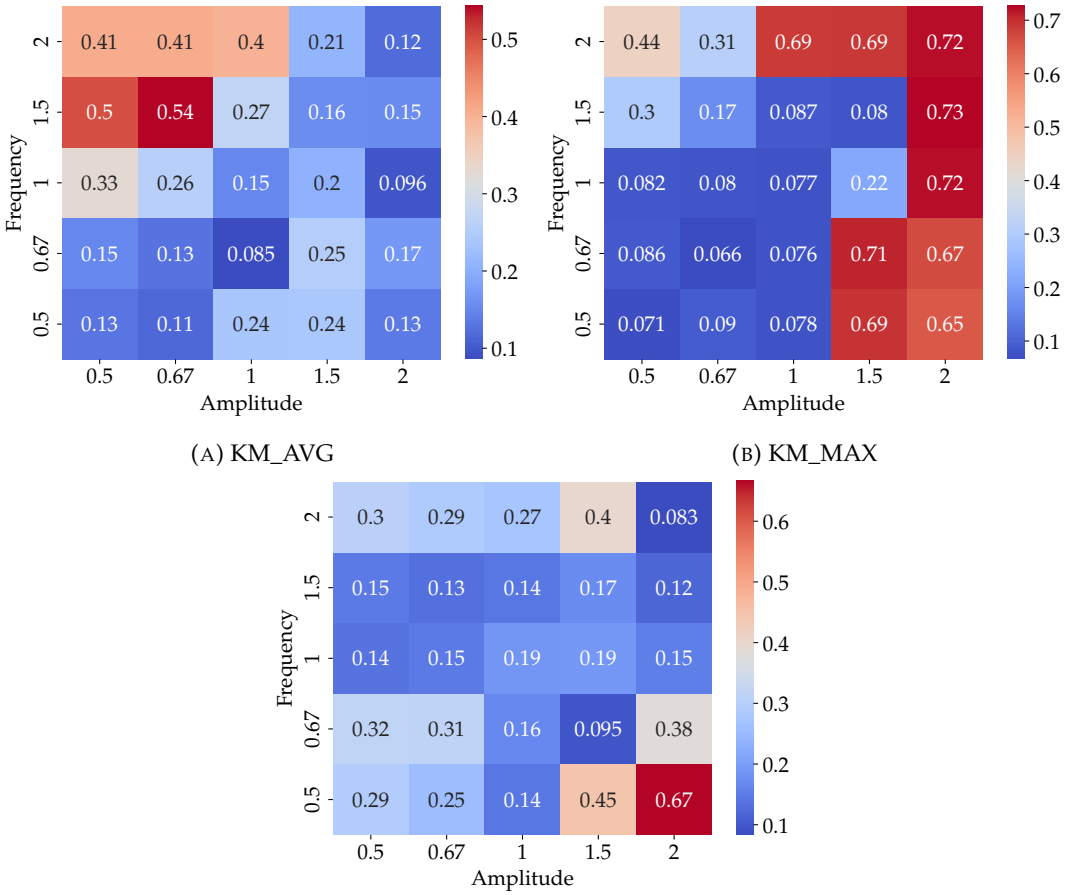
$$\alpha \cdot |t_1 - t_2| + \beta \cdot |f_1 - f_2| + \gamma \cdot |a_1 - a_2|$$

And this can be normalized such that we only have two weights α, β with respect to the time component:

$$|t_1 - t_2| + \alpha \cdot |f_1 - f_2| + \beta \cdot |a_1 - a_2|$$

Since we have no information on what could be sensible weights for our particular application, we will simply experiment with various weights. We fix the number of vertices at 300 as this seemed to provide fairly accurate estimates on all methods (see Figure 5.11 again; examples of approximations are shown in Figure B.6 of Appendix B). Considering the observation that the performance on the K-means sum variant stayed significantly behind compared to the other methods, we ignore this variant (also in favor of computational restrictions).

The average TDOA errors over the five largest peaks are shown in Figure 5.12. Our goal is to find a combination of weights that, on average, provides the best translation estimate. This appears to be different for each approximation method. We would expect that if we get closer to the optimal weight combination, that the errors also gradually decrease, but this pattern is not immediately visible. The K-means average shows a best average error of 0.085 seconds for weights $\alpha = 0.67, \beta = 1$. In this case, the frequency should have a slightly lower weight. The K-means max, however, shows areas of good and bad weights, where the good weights seem to



(A) KM_AVG

(B) KM_MAX

(C) GI

FIGURE 5.12: Hausdorff distance with weighted L_1 metric using approxmations of 300 vertices. Average TDOA errors based on ground truth (-0.799) over five largest peaks.

be mainly concentrated at $[0.5, 0.67, 1]$, with the best average error of 0.066 seconds at $\alpha = 0.67, \beta = 0.67$. This implies that the time distance should have a higher weight for this approximation type again. On the other hand, the greedy insertion requires the frequency and amplitude distances to have double the weight (i.e., $\alpha = 2, \beta = 2$) compared to the time distance with best average error of 0.083 seconds. Since the envelope representation has an average error of 0.031 seconds over the same five largest peaks, none of the approximation methods succeed in matching this performance.

Summary. Despite our efforts, it seems that spectrogram approximations are not a suitable representation to match on, as the TDOA estimates are not getting closer to the true TDOA. The approximations of the terrains may have been too coarse, or the Hausdorff distance is simply more sensitive to small deviations in this more detailed representation, which is likely the reason why we observed unstable estimates in the unweighted variants when varying the number of vertices. This is clearly less the case with the envelope representation. Considering the fact that the envelope also requires significantly less points, makes it a more suitable representation altogether. We should keep in mind that our observations are limited to a single roar (at different peaks) due to computational constraints, but we expect similar results on other roars as well and the conclusion not changing at other noise levels.

Roar	HD_PNT	HD_SEG	EMD	CR_CRR	GDOP
LP2	203.85	319.12	347.48	**66.55**	7.03
LP3-D (I)	**50.69**	96.67	83.29	56.48	6.73
LP3-D (II)	69.36	98.89	80.57	**66.23**	6.77
LP5	416.16	120.36	427.83	**107.86**	4.51
LP7	**6.71**	102.12	211.87	27.78	2.86
LP8 (I)	**248.75**	1150.49	1176.72	431.71	5.20
LP8 (II)	231.88	337.62	815.36	**93.45**	13.47
LP8 (III)	41.48	101.26	109.21	**18.93**	2.56
AVG	158.61	290.81	406.54	**108.62**	6.14

TABLE 5.5: Position errors (m) based on four recordings with highest resolution for each roar. The recording with the highest resolution was taken as reference. Best estimate for each roar is in bold.

5.3 Roar Localization

Now that we have gained more insights into the TDOA estimation performance of our methods on recordings of roars, we turn our attention to the localization performance. First we consider the localization of real roars as obtained from the field. After that, we extend the analysis to simulations of roars.

5.3.1 Real Roars

Setup. Since we need at least four microphones to unambiguously determine the position and given the fact that a roar did not always reach each microphone equally well, we pick for each roar the four microphones which have the recordings with the highest resolution. The TDOAs are then estimated with respect to the recording with the highest resolution. For each such estimation, we pick the matching that corresponds to the lowest distance from the five largest peaks. Note that our main goal is to improve the localization of the howler monkeys, which is currently best achieved using TDOA estimates from waveform (or spectrogram) cross-correlations due to their higher time resolution and richer information. We will therefore specifically compare our methods against these. Note that we do not consider the discrete Fréchet distance due to its poor scalability with little to no performance gain in terms of the TDOA estimation, as indicated in the previous section.

Results. The localization results are presented in Table 5.5. On average, the cross-correlation still seems to perform better than our methods by frequently having the best position estimates. However, the Hausdorff distance for point sets provides better estimates in three out of the eight cases. The estimate that particularly stands out is from the LP7 setup with 6.71 meters as opposed to 27.78 meters. Since we also observe a corresponding GDOP value of 2.86 that indicates good geometry of the microphones, this likely means that the Hausdorff distance provided better TDOA estimates in this case as well. Aside from this, we sometimes observe extremely large position errors up to nearly 1200 meters by the segments variant and EMD, which generally show larger errors as well. This seems to be in line with our observations made regarding the TDOA estimation, where the Hausdorff distance for point sets

Method	Mean $\pm$ SD	Median	Max	Min
HD_PNT	0.0731 ± 0.0936	0.0399	0.333	0
HD_SEG	0.104 ± 0.123	0.0554	0.551	0
EMD	0.123 ± 0.109	0.117	0.398	0
CR_CRR	0.0775 ± 0.0746	0.0558	0.317	0

TABLE 5.6: Approximation of TOA errors (s) based on four recordings with highest resolution for all roars. The recording with the highest resolution was taken as reference.

appeared to be more robust compared to the other methods. Moreover, for this method we do not observe a significant difference ($T = 12$, $p = 0.461$), whereas for both the segments variant and EMD, we do observe a significant difference ($T = 0$, $p = 0.00781$).

Measurement Errors. Ideally, we can directly relate the observed localization performance to the measurement errors of each method, but we do not immediately have access to the ground truths. However, we can approximate them using the measured two-dimensional position of the roar, the positions of the microphones and our speed of sound assumption. This only ignores the heights, which still results in fairly accurate positions with an average error of around 17 meters (see Table B.6 of Appendix B).

The TOA errors for each method can be found in Table 5.6. Note that we specifically refer to TOAs instead of TDOAs, since the localization algorithm expects TOAs as input. The TDOAs are relativized to become pseudo-TOAs with the microphone that is first reached having a TOA of zero, and after that corrected for synchronization. We observe approximately equal errors for both the Hausdorff distance for point sets and the cross-correlation method, slightly larger errors for the segments variant and subsequently slightly larger errors for the EMD. This seems to be reflected in the localization performance. Note that the cross-correlation TDOAs have been derived from the waveform representation with a time resolution that is $0.128/(1/2000) = 256$ times more precise than our spectral energy envelope and that has additional phase information of which it could benefit, so it should be capable of more exact estimation in those respects. However, the Hausdorff distance for point sets still seems to match its performance surprisingly. The findings should still be interpreted with caution given the relatively small sample size.

Other Factors. An average position error of more than 100 meters is still considered relatively high. Other factors than the TDOA estimates that have likely affected the accuracy are:

- Poor root choice.

  If we look at Table B.8 of Appendix B, we observe that especially the cross-correlation suffered from this, which might indicate that more precise TDOAs result in root solutions with similar sum of squares discrepancies, increasing the likelihood of making the wrong choice. For the other methods, it mostly picked the right one, but we also observe no changes in more than half of their estimates. If the localization algorithm obtains a negative discriminant when solving the equation, it sets this to zero to still obtain a location, so the positive

Roar	HD_PNT	HD_SEG	EMD	CR_CRR
LP2	203.85	319.12	347.48	65.02
LP3-D (I)	50.69	72.74	83.29	56.48
LP3-D (II)	69.36	18.30	123.55	68.13
LP5	153.80	285.23	143.04	10.81
LP7	6.71	102.12	211.87	27.78
LP8 (I)	145.42	327.42	627.20	267.84
LP8 (II)	351.33	206.93	254.89	200.55
LP8 (III)	55.56	185.94	109.21	19.57
AVG	129.59	189.72	237.56	89.52

TABLE 5.7: Position errors (m) based on all combinations of five recordings with highest resolution for each roar. The recording with the highest resolution was taken as reference. Improvement is marked green and decline is marked red with respect to Table 5.5.

and negative roots represent the same solution in that case. This means that, for the most part, the algorithm was not able to find a proper solution that best matches the set of TDOAs, which is often times an indication of large measurement errors.

- Inaccurate measurements of the microphone and roar positions.
- Inaccurate time synchronization.

  Measurements in clock deviations are made during the start and end of the deployment, which may be slightly off, and the assumption that drift occurs in a linear pattern may also not completely hold in practice.
- Inaccurate speed of sound.

  We had no measurements of the exact temperature during each of the recordings and therefore assumed a temperature of 26° C. This affects the solving process of the localization algorithm to find an estimate position that best matches the TDOAs, as these depend on the speed of sound in that environment at that moment.
- Suboptimal microphone placement.

  With an average GDOP value of around 6, the geometry of the microphones gets a moderate rating and has a noticeable impact on the accuracy with slight deviations in the TDOA measurements.

Improvements. Having the same set of recordings at each roar to determine the position allowed for a more fair comparison between the methods. This also came with the benefit of being able to analyze and compare the GDOPs and TOA errors. Most of the times, the position estimates can be improved by estimating more TDOAs and trying out different combinations of those to optimize a predicted position error (calculated by the Sound Finder software). The results can be found in Table 5.7. Note that the results concerning the roar from the LP7 setup are identical, which is due to

a lack of synchronization measurements on one of the recorders. It shows on average improvements for all methods, but the ranks remain unchanged. This also does not affect our earlier made conclusions, since the Hausdorff distance for point sets still shows no significant difference ($T = 10$, $p = 0.3125$), whereas the segments variant ($T = 3$, $p = 0.0390625$) and the EMD ($T = 0$, $p = 0.0078125$) still show a significant difference.

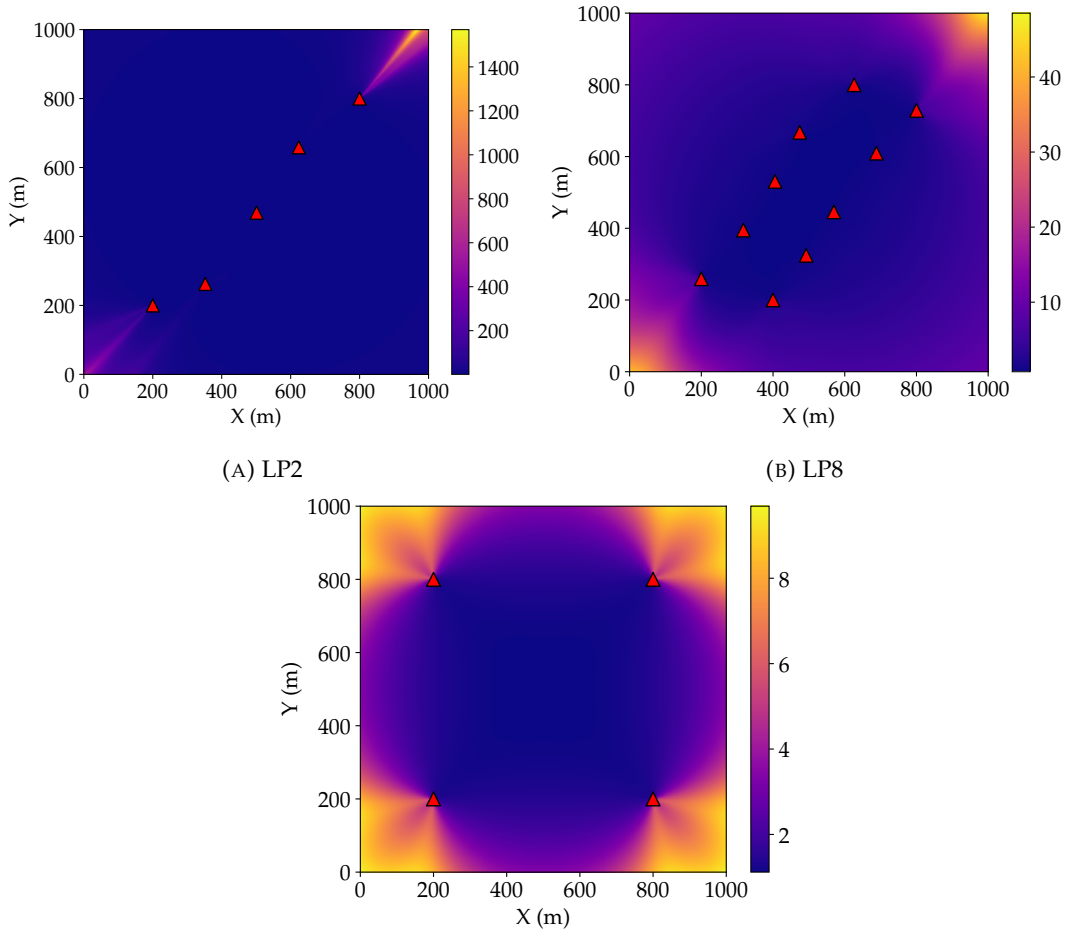
So far, we always took the roar with the highest resolution as reference, but this might not always be the best template to match against. One could think that a roar with a slightly lower resolution allows for a better matching as the discrepancy between the template and target decreases (i.e., the discrepancy between the highest and lowest resolution is expected to be larger). Therefore, it may be beneficial to additionally try out different combinations of references, which requires separate TDOA estimates for each reference combination. The results can be found in Table B.9 of Appendix B, which shows improvements on some estimates, but on average performs worse on our methods. The predicted position error used as optimization criterion significantly deviates from our ground truths, which again causes the algorithm to pick the wrong estimate. This could potentially be further improved if we only consider combinations of which we are more certain that the TDOAs are accurate, for example those that involve recordings with a high resolution or low spectral entropy (which tells something about the spread of the energy), but this requires a more advanced analysis.

5.3.2 Simulated Roars

While the localization of real roars from the field gives us an idea how our methods perform in practice, it is difficult to explain some of the observed outcomes. In particular, we expected the cross-correlation to have a significant advantage over the other methods, which did not always appear to be the case. To obtain more insight in this, we perform a simulation of roars. Note that our goal here is not to create a challenging setting with respect to TDOA estimation, but rather investigate the limitations of our envelope representation and the effects of the microphone geometry on those for each method.

Microphone Setups. We will consider three different microphone setups as shown in Figure 5.13, which additionally displays the GDOP value at each position. Two of those setups are directly taken from the real dataset and projected onto our allowed microphone range. From the five setups in the real dataset, four shared the same collinear structure consisting of five microphones and the remaining setup takes two of those next to each other, so these two setups are the most distinct. The other setup we consider puts the microphones in the four corners, forming a square, because of its beneficial geometry with well-defined areas. If we look at their GDOP values, we observe that within its convex hull, the geometry has the least impact, but in the corners it gets a significant larger impact. If the position lies on the line formed by the microphones of the LP2 setup for example, the accuracy becomes extremely poor. This concretely means that if a howler monkey would roar in this area and we would be trying to estimate its position, the TDOAs need to be (nearly) exact to be any useful.

Perfect Geometry. We start with the simulation in the region where the impact of the geometry is at a minimum, which means that the localization accuracy is largely determined by the signal representation and the characteristics of the TDOA

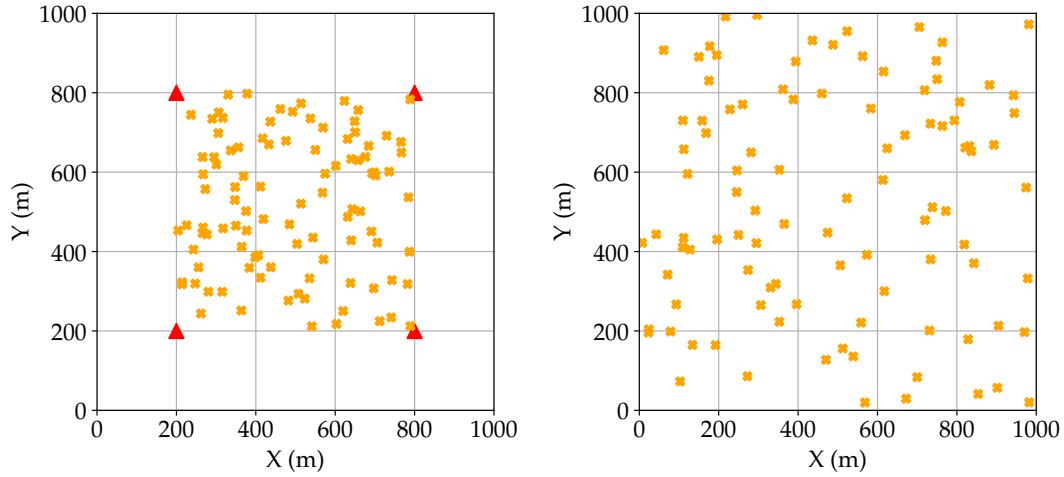


(A) LP2

(B) LP8

(C) Square

FIGURE 5.13: Custom setups or projections of setups from our real dataset onto the simulation grid and their GDOP values. The red triangles denote the microphones.



(A) Within convex hull of square setup.

(B) Over entire grid (in any setup).

FIGURE 5.14: Location samples in the simulation grid. The red triangles denote the microphones and the orange crosses denote the samples.

Method	Mean $\pm$ SD	Median	Max	Min
HD_PNT	28.97 ± 62.03	14.49	406.19	1.35
HD_SEG	13.00 ± 39.87	4.34	287.57	0.19
EMD	40.50 ± 94.00	16.09	562.34	1.35
CR_CRR (env)	46.31 ± 103.37	15.64	512.73	1.35
CR_CRR (wf)	0.055 ± 0.028	0.053	0.15	0.0017
GDOP	1.23 ± 0.089	1.21	1.46	1.12

TABLE 5.8: Position errors (m) of 100 positions sampled within the convex hull of the square setup.

Method	Mean $\pm$ SD	Median	Max	Min
HD_PNT	0.0321 ± 0.0193	0.0315	0.0723	0.000161
HD_SEG	0.0105 ± 0.0102	0.00702	0.0481	$3.39 \cdot 10^{-6}$
EMD	0.0329 ± 0.0211	0.0320	0.157	0.000161
CR_CRR (env)	0.0318 ± 0.0187	0.0315	0.0658	0.000161
CR_CRR (wf)	$0.000134 \pm 9.59 \cdot 10^{-5}$	0.000131	0.000893	$2.70 \cdot 10^{-6}$

TABLE 5.9: TDOA errors (s) of 100 positions sampled within the convex hull of the square setup.

estimation method being used. For this, we uniformly sample 100 positions within the convex hull of the square setup (see Figure 5.14a). For the TDOA estimation, we always take the closest microphone as reference. Note that in this case, the simulation setup is equivalent to only sampling positions in the Voronoi region within the hull for some reference microphone due to the symmetry of the microphone placements. Since we expect the quality of the TDOAs to be independent from the choice of reference here, this has (little to) no impact on the localization performance and is only done for simplicity purposes.

The localization results are presented in Table 5.8. Note that for the cross-correlation method, we applied TDOA estimation to both the envelope and waveform representation. Our first observation is that the GDOP values are in the range $1 - 2$, indicating excellent precision in terms of geometry as expected. Despite the relatively easy TDOA estimation setup, we still observe differences in the localization performance. The Hausdorff distance for point sets seems to perform on average slightly better than the EMD and cross-correlation for envelopes which share similar performances, but their median errors are fairly similar with 15 meters, so it may occasionally find a slightly better TDOA estimate that still has a noticeable impact on some position estimates. This seems to be in line with our earlier observation where it is more robust in lower noise level settings.

If we take a look at the TDOA errors in Table 5.9, the errors between those methods all have an average (and median) of around 0.032 seconds. Given the time resolution of 0.128 seconds and the observation that the algorithms find their optimal matching by aligning two points, this means that the errors will generally lie in the range $[0, 0.064]$. Since the positions are uniformly sampled, the TDOAs are consequently also uniformly sampled, so the errors are on expectation 0.032 seconds, matching our observation.

Setup	HD_PNT	EMD	CR_CRR (env)	CR_CRR (wf)	GDOP
LP2	133.34	138.01	142.33	0.41	4.72
LP8	38.55	37.82	34.40	0.18	3.53
Square	19.43	20.96	20.66	0.090	1.93

TABLE 5.10: Median position errors (m) of 100 positions sampled over the entire grid for multiple setups.

The Hausdorff distance for segments does not have this characteristic and should therefore be less restricted by the time resolution, which is reflected in the TDOA estimates with an average error of around 0.010 seconds and median position error of 4 meters, both being three times more accurate. This is different from the localization of real roars, where this variant performed on average worse than the point sets, which can be explained by the easier TDOA setup causing less fluctuating estimates. If we compare the cross-correlation methods on the two different representations, we observe a noticeable difference in performance where the waveform allows for TDOA estimates of more than 200 times the precision. This also translates to the position estimates. Given the relatively easy TDOA setup, the phase information in the waveform has a negligible contribution to the quality of the estimates, so this clearly shows the time resolution being the largest bottleneck. However, we now observe a significant performance difference between the Hausdorff distance for point sets and the waveform cross-correlation, which was not the case with the localization of real roars. In practice, this effect might therefore be smaller.

Effect of Geometry. Now that we have gained more insight into the limitations of the envelope representation on the TDOA estimates specifically, we will investigate how the geometry of the microphones amplify these errors in the position estimate. We uniformly sample 100 positions over the entire grid (see Figure 5.14b) and try to estimate them for each of our three different setups.

For conciseness and because of the large variance in the position estimates, only the median results are shown in Table 5.10 (the full tables can be found in Appendix B). The Hausdorff distance for segments is omitted due to computational restrictions, but we expect that the observations we make on the other methods similarly apply there. As the TDOA errors are (nearly) identical as well, we will ignore the analysis of those. Let us first focus on the methods applied to the envelope representation. We observe that the collinear setup of LP2 is suboptimal with position estimates that are seven times less accurate compared to the square setup. An additional line-up in the LP8 setup improves this significantly, but is still twice less accurate. This is also reflected in their GDOP values. If we now compare this to the cross-correlation method applied to waveforms, we observe approximately the same proportions, but the absolute differences are clearly on a significant lower scale. Depending on the desired precision of the position estimates, the geometry of the microphones becomes more important if we would be considering a representation with a lower time resolution such as our envelope.

Chapter 6

Conclusion

In this thesis, we investigated whether taking a geometric approach for estimating the time difference of arrival between two sound signals improves the localization of howler monkeys. The state-of-the-art for obtaining these differences is by cross-correlation, which has been unsuccessful in accurately estimating them for signals that are significantly obscured by noise. This consequently has a negative impact on the precision of the position estimates. Since the signal data can be treated as geometric entities that form shapes in space, the goal was therefore to find more robust methods in the area of computational geometry.

For this, we modeled the problem of shape matching under one-dimensional translations, which is related to the problem of TDOA estimation, and proposed several methods for solving this. We obtained a general exact algorithm for the Hausdorff distance and applied this to point sets, line segments in the plane, and triangles in three-dimensional space, that all run in $O(mn \log(m+n))$ time. We adapted methods for the Fréchet distance and obtained a $(1+\varepsilon)$-approximation for both the continuous and discrete variant that run in $O((mn)^2(m+n)\log(1/\varepsilon))$ and $O(m^2n^2\log(1/\varepsilon))$ time, respectively. Finally, we introduced an exact algorithm for the Earth Mover's Distance when the underlying metrics are L_1 or L_∞, and a 2-approximation for the other norms. This requires $O(mn)$ times solving the static distance (using any algorithm).

A first indication of the noise robustness of each of our proposed methods was given by matching a clean simple signal (sine waves or white noise signal) with a shifted noisy copy. This showed that the Earth Mover's Distance was most robust in any case. For the Hausdorff distance methods, the segments variant appeared to be more robust in higher noise levels, especially when the signal is more structured, but the point sets variant with underlying metric L_1 performed overall better in the lower noise levels.

In the more practical setting with roars as signal type, we focused on a spectral energy envelope representation of the signal to cope with the poor scalability of our methods. The cross-correlation showed to be significantly more robust than our methods. Attempts to improve the estimation quality further by considering spectrogram approximations of the signals have not been successful either. Contrary to expectations, the localization of eight roars showed no significant difference between the cross-correlation and Hausdorff distance for point sets, whereas the segments variant and Earth Mover's Distance still performed significantly worse. Simulations showed the impact of the limitations of our current signal representation on the precision of the position estimates. This moreover proved the importance of the microphone geometry, especially when dealing with these limitations.

Based on these observations, we have to conclude that a geometric approach for TDOA estimation does not improve the localization of howler monkeys. The largest bottleneck appeared to be the scalability of our methods, which is why we

had to rely on simplifications of the original signal representation and were not able to use all methods effectively (i.e., the Fréchet distance). An interesting direction for future research would be to design and try out more efficient geometric methods, perhaps other approximations or heuristics, that are capable of handling larger time resolutions. One potential approach that could be worth looking into is to increase the time resolution and use a similar optimization method as cross-correlation, where we slide the template over the target and compute at each step the static distance, which has a better scalability. This would result in suboptimal matchings between the template and target, but the increased time resolution partially compensates for this. Another more interesting approach would be to reduce the number of segment matchings. We now matched a template to each possible segment of the target, but we may be able to eliminate segments that do not match the frequency content of the template for example, or we could simply focus first on the peaks of the target, since the template is also selected based on the largest peaks, and terminate earlier if the matching quality only decreases at a certain point. Although a larger time resolution is expected to improve the TDOA estimation, it could also result in a signal representation that is more oscillatory (or noisy) again due to the time-frequency trade-off to which the geometric methods could be more sensitive, so it might have the opposite effect. Other than those directions, one could try out several noise reduction techniques (e.g., Wiener filtering or spectral subtraction), which may have a positive effect on the matching performance.

With respect to the localization itself, related work [25, 66] showed mean position errors of 27 and 58 meters for larger sample sizes and with manually verified TDOAs. In this study, the cross-correlation showed the best performance with a mean error of 90 meters, performing slightly worse in comparison. The simulation indicates that the current microphone setup used to capture the roars may have been suboptimal, so we expect the performance to improve with setups that have a more beneficial geometry (e.g., a square or star shape). One other factor that had a noticeable impact on the performance was the algorithm being used, as it picked the wrong root solution in some cases. It currently selected the solution with the least sum of squares discrepancy, but one could also use a different criterion (e.g., the pseudorange error which appeared to work better in some test cases [52]). Alternatively, it may be beneficial to look at more robust localization techniques, either in the iterative (e.g., hyperbolic least squares) or non-iterative (e.g., maximum likelihood estimator) category [33].

Appendix A

Additional Background

This chapter contains additional background information of the techniques being used in the context of this research.

Bancroft Algorithm

For a microphone i, we have the vector:

$$p_i = \begin{bmatrix} X_i \\ Y_i \\ -c \cdot \tau_{i,1} \end{bmatrix}$$

The Lorenz inner product between two vectors p_i and p_j is calculated by:

$$\langle p_i, p_j \rangle = X_i \cdot X_j + Y_i \cdot Y_j - c^2 \cdot \tau_{i,1} \cdot \tau_{j,1}$$

We define the matrix:

$$B = (p_1, p_2, ..., p_M)^T = \begin{bmatrix} x_1 & y_1 & -c \cdot \tau_{1,1} \\ x_2 & y_2 & -c \cdot \tau_{2,1} \\ \vdots & \vdots & \vdots \\ x_M & y_M & -c \cdot \tau_{M,1} \end{bmatrix}$$

We need the vectors u and v that solve the linear systems:

$$Bu = e \qquad Bv = a$$

where e is the all-ones vector of length M and a is defined as:

$$e = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \qquad a = \frac{1}{2} \begin{bmatrix} \langle p_1, p_1 \rangle \\ \langle p_2, p_2 \rangle \\ \vdots \\ \langle p_M, p_M \rangle \end{bmatrix}$$

For a setup consisting of $M = 3$ microphones, we can easily solve it using the inverse of B (i.e., $u = B^{-1}e$ and $v = B^{-1}a$) since it is a square matrix. For a setup with $M > 3$ microphones, which results in an overdetermined system and therefore requires a least-squares solution, the pseudo-inverse of B is used (through, for example, QR-decomposition).

We then have the scalar coefficients:

$$E = \langle u, u \rangle$$
$$F = \langle u, v \rangle - 1$$
$$G = \langle v, v \rangle$$

which are then used in the final equation:

$$E \cdot \lambda^2 + 2F \cdot \lambda + G = 0$$

This gives us two solutions: a positive root λ^+ and a negative root λ^-. The corresponding source locations are calculated by:

$$q^+ = \lambda^+ u + v = \begin{bmatrix} x^+ \\ y^+ \\ c \cdot \tau^+ \end{bmatrix} \qquad q^- = \lambda^- u + v = \begin{bmatrix} x^- \\ y^- \\ c \cdot \tau^- \end{bmatrix}$$

where τ^+ and τ^- represent the time of emission from the source relative to time delays $\tau_{i,1}$ for $i = 1, ..., M$ where $\tau_{1,1} = 0$ (which means that we expect those to be negative in general). For more details of its derivation, we refer to the original paper [12].

Weighted K-means

We describe the algorithm as implemented by the *scikit-learn* library [60]. Suppose we have a set of N points in the grid and we want to reduce it to $K < N$ clusters. Each point $x_i = (t_i, f_i)$, where t_i and f_i represent the corresponding time and frequency bins, is assigned a weight w_i which corresponds to the amplitude value. The objective is then to minimize the sum of weighted squared distances between the points and their assigned cluster centers:

$$\sum_{k=1}^{K} \sum_{x_i \in C_k} w_i \cdot d(x_i, \mu_k)$$

where μ_k is the centroid of the cluster C_k and $d(x_i, \mu_k)$ is the Euclidean distance between x_i and μ_k. This is also known as the inertia score, which measures how well the data is clustered. Points with higher weights will pull the centroid closer to them and therefore regions with higher amplitude values are more highlighted. The algorithm first randomly initializes K centroids. Then an assignment step occurs where each point x_i is assigned to the nearest cluster based on the regular Euclidean distance. After that, an update step occurs where the centroids are updated based on the weighted mean of the points:

$$\mu_k = \frac{\sum_{x_i \in C_k} w_i \cdot x_i}{\sum_{x_i \in C_k} w_i}$$

These last two steps are repeated until convergence. In practice, this algorithm runs relatively fast, but it is prone to falling in local minima. This is why the algorithm is often run for multiple iterations and the iteration with the lowest inertia score is selected. After the clustering, we must assign amplitude values to the cluster centers to obtain the approximation of the spectrogram, which can be achieved by aggregating the weights of the points that belong to the same cluster.

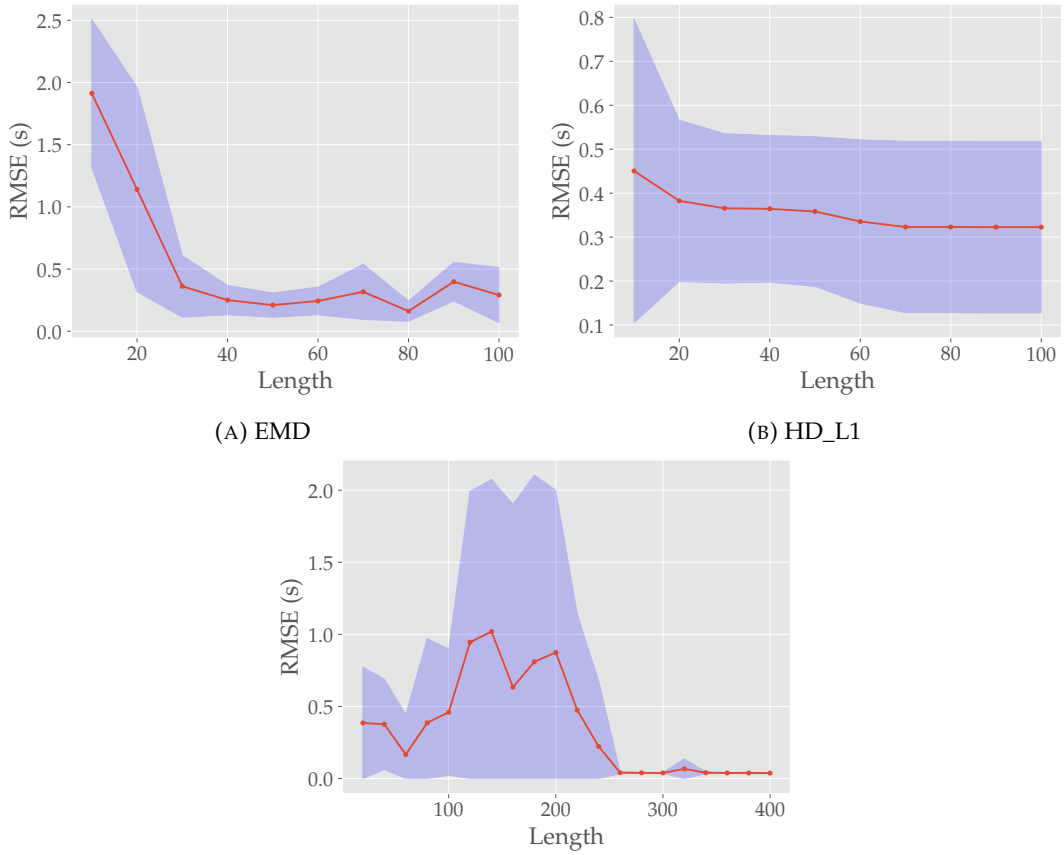
Greedy Insertion

If we view the spectrogram as a terrain (or height field), we can try to approximate it with a mesh of triangles, often called a triangulated irregular network (TIN), by minimizing the number of points. One simple algorithm that efficiently achieves this goal is greedy insertion [34], which in contrast to the weighted K-means method, samples important points instead of aggregating important regions. It starts with an initial approximation of two triangles by picking the four corner points of the grid. Then it repeatedly searches for the unused point with the largest error and adds it to the current approximation. The vertical error of a point is often used as (local) error measure, which is obtained by taking the difference between the actual height and the interpolated height of the approximation. This procedure terminates when a certain condition is met, for example, the error (which may be a more global measure such as the sum of vertical errors) is below a threshold, or a maximum number of points or triangles is reached. The triangulation that results from this satisfies the Delaunay condition.

Appendix B

Additional Results

Effect of Template Length

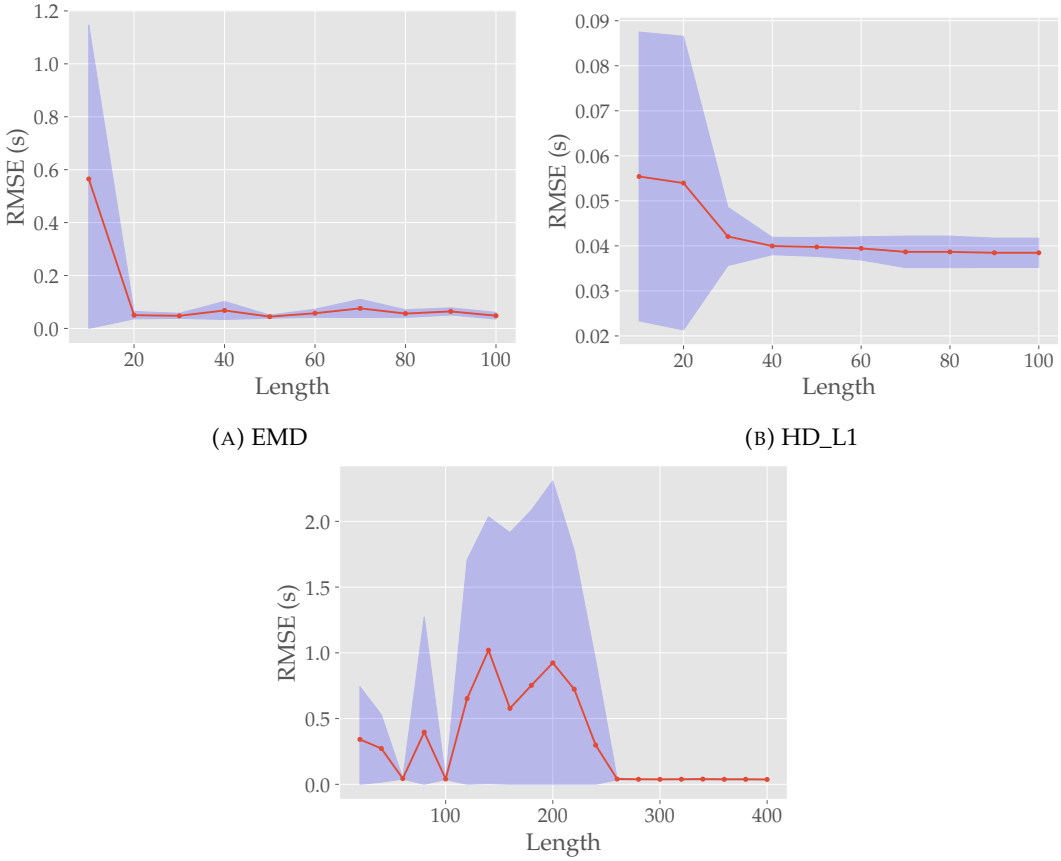


(A) EMD

(B) HD_L1

(C) CR_CRR

FIGURE B.1: Effect of the template length on the matching performance (SNR = 0dB and $\alpha = 0.6$). The red line shows the mean and the blue area shows the 95% confidence interval based on the RMSE of five peaks. The RMSE of a single peak is determined over 100 instances.



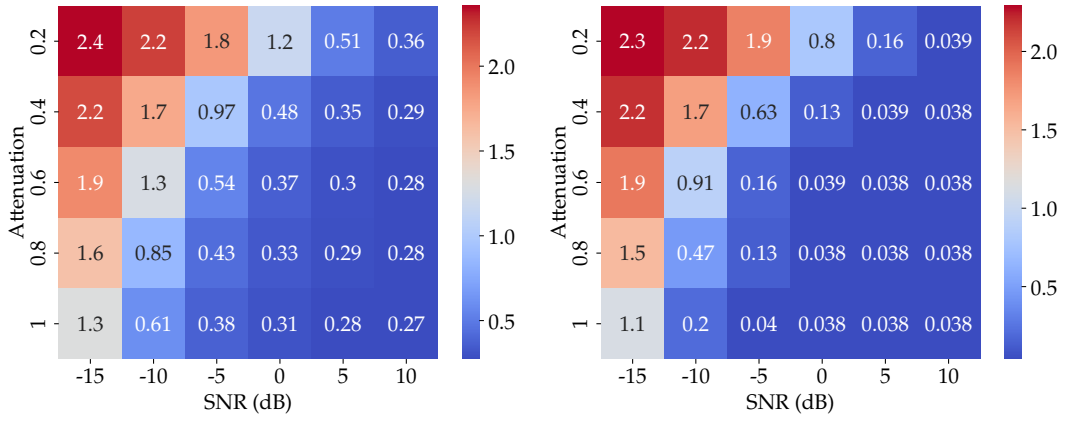
(A) EMD

(B) HD_L1

(C) CR_CRR

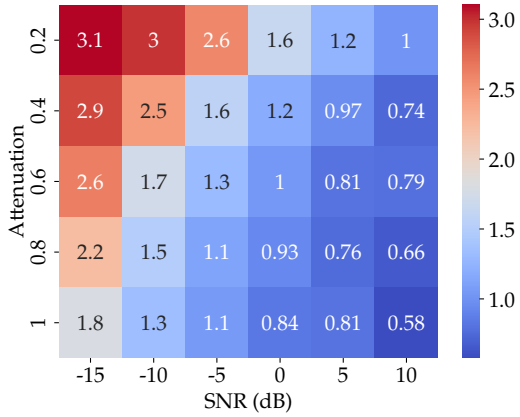
FIGURE B.2: Effect of the template length on the matching performance (SNR = 10dB and $\alpha = 1$). The red line shows the mean and the blue area shows the 95% confidence interval based on the RMSE of five peaks. The RMSE of a single peak is determined over 100 instances.

Noise Robustness Analysis



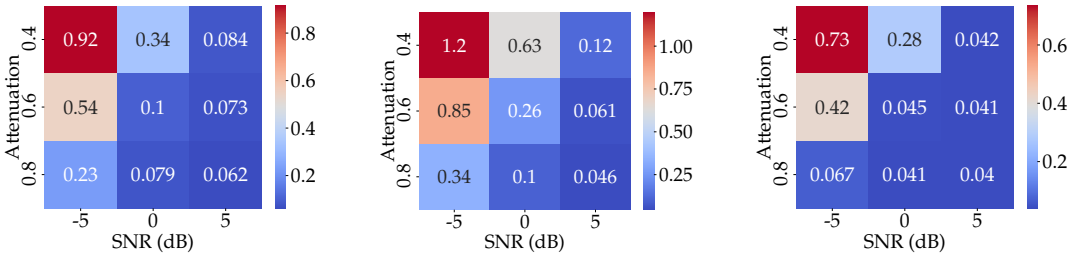
(A) CR_CRR (40)

(B) CR_CRR (300)



(C) EMD (300)

FIGURE B.3: Noise robustness on the roar with the highest resolution for different template lengths. Average RMSE values over the five largest peaks. The RMSE of a single peak is determined over 100 instances.



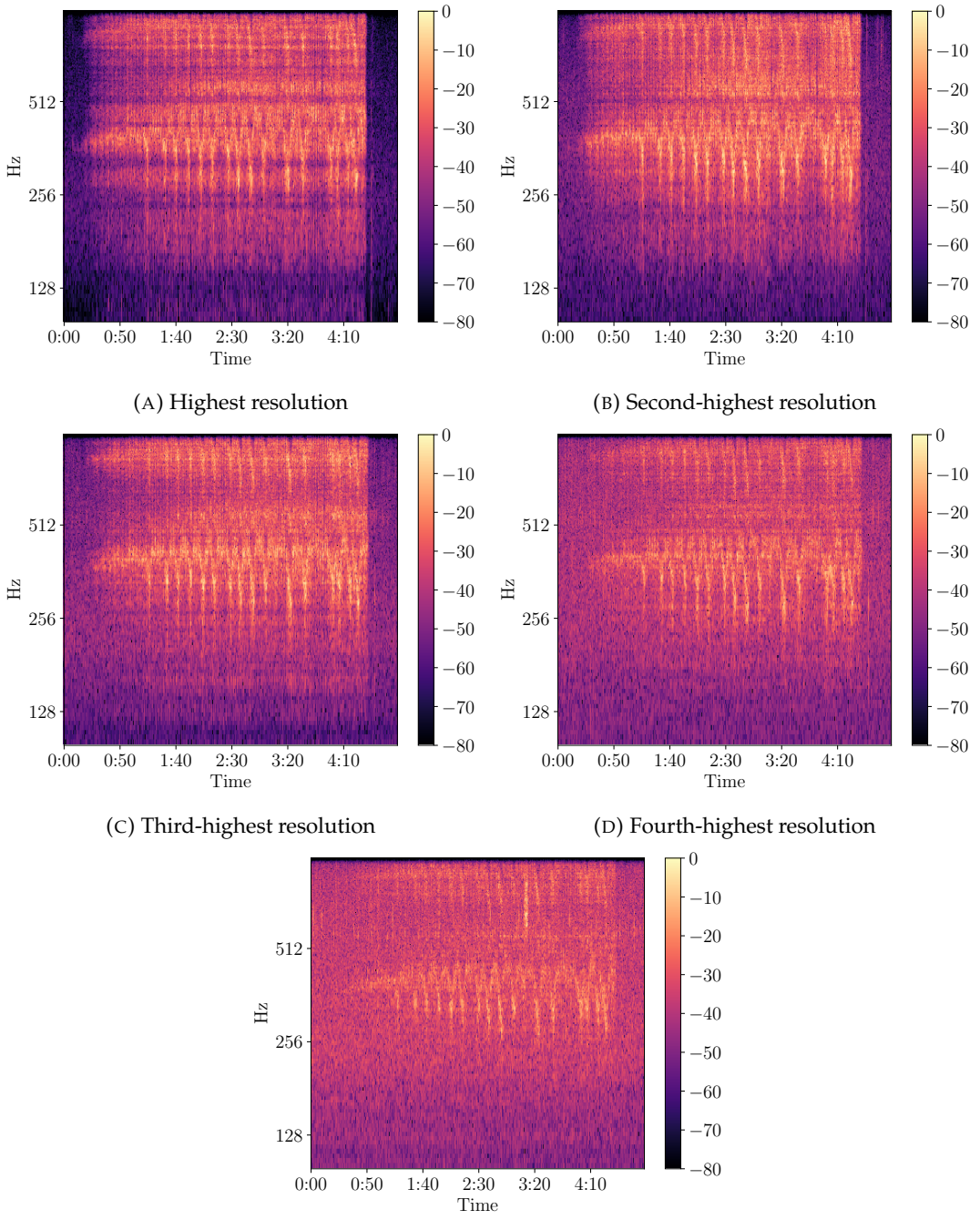
(A) EMD

(B) HD_L1

(C) CR_CRR

FIGURE B.4: Noise robustness on the roar with the highest resolution and template length 100. RMSE over 100 instances at the largest peak.

Real Estimation



(A) Highest resolution

(B) Second-highest resolution

(C) Third-highest resolution

(D) Fourth-highest resolution

(E) Fifth-highest resolution

FIGURE B.5: Same roar captured by multiple microphones and ranked according to their RMS value (LP3-D (I)).

ε	dist	t	t_error	run_time
0.1	0.5332	-0.4958	0.3032	34m
0.01	0.4951	-0.7487	0.0503	1h2m
0.001	0.4927	-0.7497	0.0493	1h43m
0.0001	0.4925	-0.7498	0.0492	2h23m

TABLE B.1: Results of discrete Fréchet distance at largest peak for different thresholds (true is -0.799).

Peak	HD_PNT		HD_SEG		DIS_FD ($\varepsilon = 0.01$)		EMD	
	t	dist	t	dist	t	dist	t	dist
1	0.0	0.318	$-$**0.0241**	0.188	-0.117	0.457	-0.128	0.285
2	-0.128	**0.263**	-0.115	**0.164**	**0.00252**	0.419	-0.128	**0.154**
3	**0.0**	0.418	0.0166	0.330	-0.122	0.448	-0.256	0.379
4	-0.128	0.343	0.0421	0.212	$-$**0.113**	**0.362**	-0.128	0.362
5	0.0	0.362	$-$**0.0393**	0.271	-0.125	0.371	-0.256	0.204

TABLE B.2: Top-5 largest peaks matchings for a single roar (LP3-D (II)) between two high-resolution recordings. For each peak, the best estimation compared to true TDOA (-0.0468) is in bold. For each algorithm, the smallest distance is in bold.

Peak	HD_PNT		HD_SEG		DIS_FD ($\varepsilon = 0.01$)		EMD	
	t	dist	t	dist	t	dist	t	dist
1	-1.024	0.795	$-$**1.170**	0.583	-1.012	0.800	-1.152	0.280
2	-1.664	**0.648**	-1.601	0.562	-1.650	**0.657**	$-$**1.536**	0.323
3	1.664	0.785	1.664	0.635	1.577	0.790	$-$**1.408**	0.412
4	-1.408	0.657	$-$**1.232**	**0.541**	-1.383	0.686	-1.536	**0.233**
5	-1.664	0.765	-2.300	0.637	$-$**1.652**	0.771	-2.304	0.335

TABLE B.3: Top-5 largest peaks matchings for a single roar (LP3-D (II)) between a high-resolution and low-resolution recording. For each peak, the best estimation compared to true TDOA (-1.292) is in bold. For each algorithm, the smallest distance is in bold.

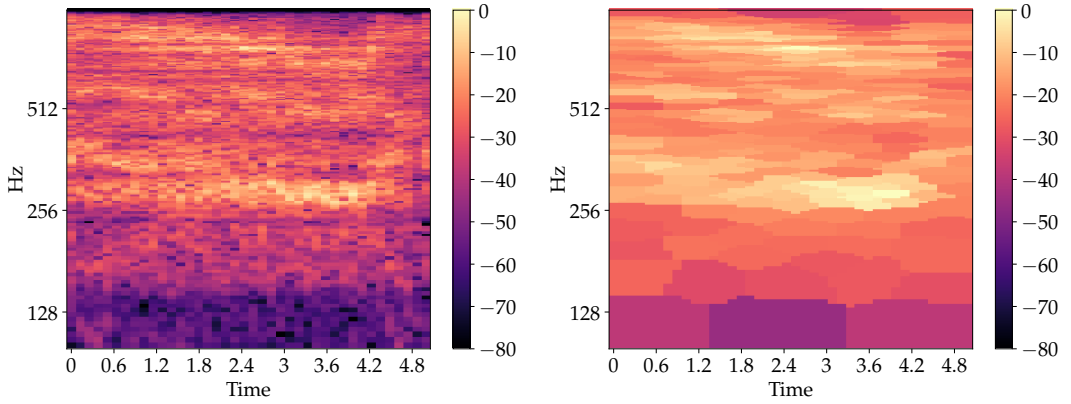
Spectrogram Approximations

vertices	t	dist	template_error	target_error	run_time
40	-0.375	0.997	7.608	6.450	1h31m
60	-0.616	1.313	6.509	6.678	3h43m
80	-0.768	1.563	6.046	5.315	6h42m

TABLE B.4: Hausdorff distance for triangles using a greedy insertion approximation (baseline is -0.768 and true is -0.799). Results are from the largest peak.

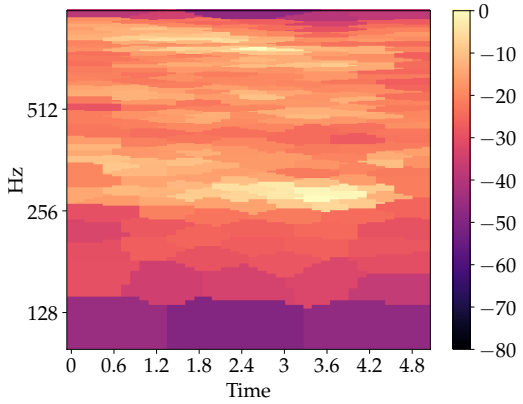
Peak	Mean $\pm$ SD	Median	Max	Min
1	1.048 ± 1.3473	0.641	17.318	$9.163 \cdot 10^{-6}$
2	1.013 ± 1.390	0.624	22.999	$2.167 \cdot 10^{-6}$
3	0.917 ± 1.070	0.565	12.612	$2.373 \cdot 10^{-6}$
4	0.962 ± 1.218	0.576	15.566	$2.163 \cdot 10^{-6}$
5	0.984 ± 1.121	0.633	12.786	$1.255 \cdot 10^{-6}$

TABLE B.5: Descriptive statistics of the amplitudes for the templates at the roar with the highest resolution.

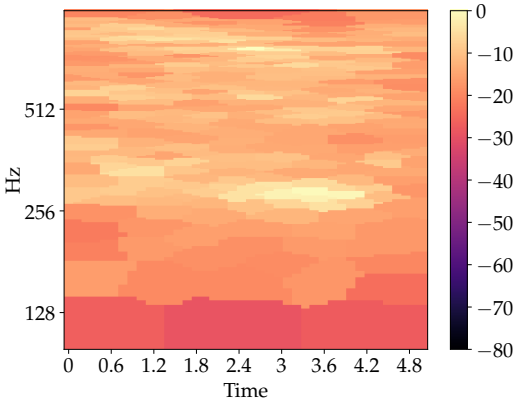


(A) Normal (9240 points)

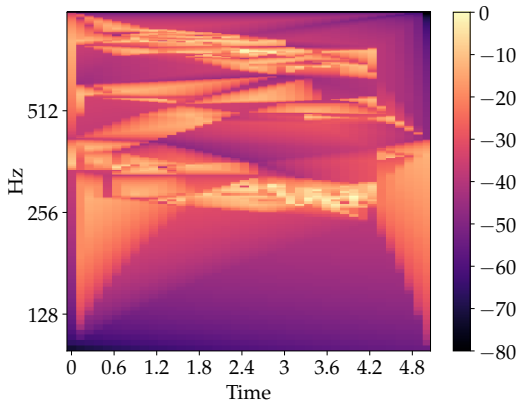
(B) K-means (max)



(C) K-means (avg)

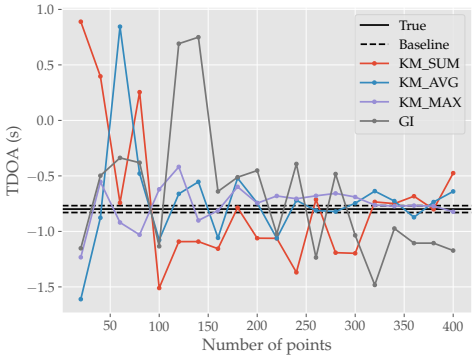


(D) K-means (sum)

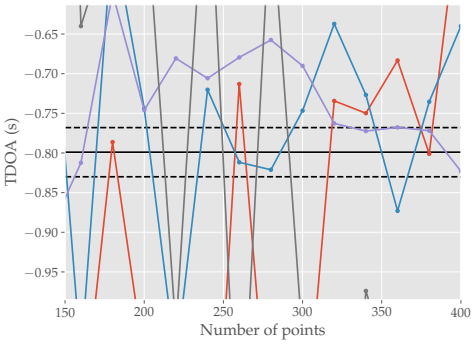


(E) Greedy Insertion

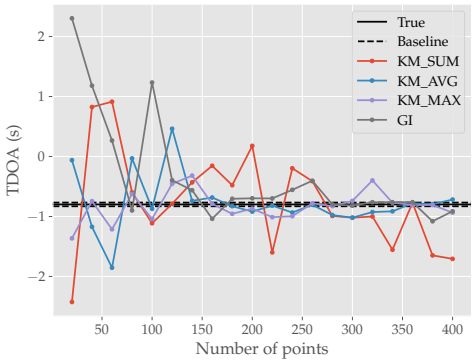
FIGURE B.6: Terrain approximations at 300 points for a segment of the roar with the highest resolution at the largest peak.



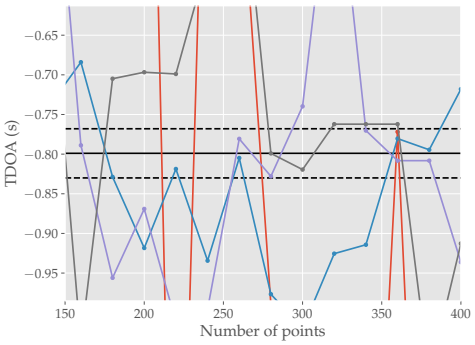
(A) Peak 2



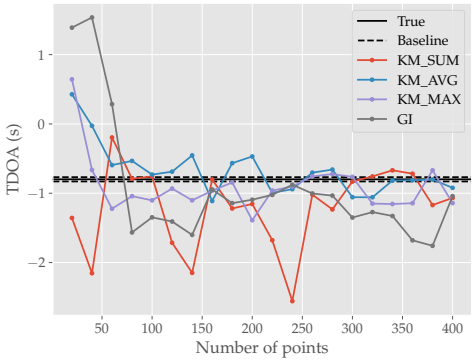
(B) Peak 2 (zoomed in)



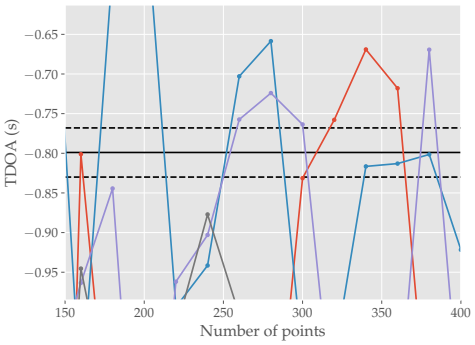
(C) Peak 3



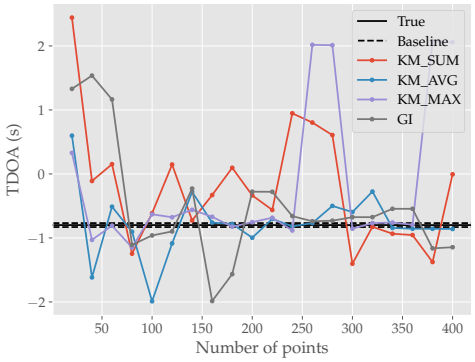
(D) Peak 3 (zoomed in)



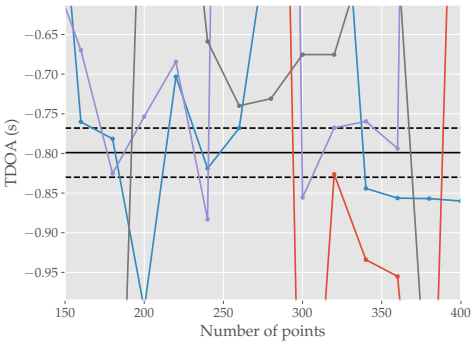
(E) Peak 4



(F) Peak 4 (zoomed in)



(G) Peak 5



(H) Peak 5 (zoomed in)

FIGURE B.7: Estimated TDOA on spectrogram approximations between two high-resolution recordings for the Hausdorff distance (L_1).

Roar Localization

Roar	low_discr	high_discr
LP2	12.36	12.36
LP3-D (I)	21.55	18.19
LP3-D (II)	16.81	26.09
LP5	1.83	1.83
LP7	65.19	0.059
LP8 (I)	0.23	482.20
LP8 (II)	2.67	1.64
LP8 (III)	16.24	15.31
AVG	17.11	69.71

TABLE B.6: Position errors (m) based on four recordings with highest resolution for each roar. Localization based on hypothetical TOAs of known two-dimensional position. Both solutions of the lower and higher discrepancy are given.

Roar	Top-4	Top-5
LP2	72.30	328.74
LP3-D (I)	50.69	50.69
LP3-D (II)	70.44	108.37
LP5	416.16	153.80
LP7	163.06	163.06
LP8 (I)	365.01	233.41
LP8 (II)	291.93	274.81
LP8 (III)	183.26	35.32
AVG	201.61	168.53

TABLE B.7: Position errors (m) for cross-correlation applied to envelope representation. Top-4 shows results based on four recordings with highest resolution for each roar, and top-5 shows results based on all combinations of five recordings. The recording with the highest resolution was taken as reference. Improvement is marked green and decline is marked red with respect to Top-4.

Roar	HD_PNT	HD_SEG	EMD	CR_CRR
LP2	266.41	319.12	623.46	65.02
LP3-D (I)	50.69	96.67	87.60	55.33
LP3-D (II)	69.36	98.89	80.57	66.23
LP5	416.16	155.00	424.31	107.86
LP7	50.69	462.83	211.87	300.89
LP8 (I)	248.75	1150.49	1176.72	119.42
LP8 (II)	208.97	338.49	815.36	93.45
LP8 (III)	55.56	101.26	110.99	0.33
AVG	170.82	340.34	441.36	101.07

TABLE B.8: Position errors (m) based on four recordings with highest resolution for each roar. The recording with the highest resolution was taken as reference. Solutions are from the higher discrepancy. Improvement is marked green and decline is marked red with respect to Table 5.5.

Roar	HD_PNT	HD_SEG	EMD	CR_CRR
LP2	203.85	105.26	347.48	60.19
LP3-D (I)	40.68	72.57	118.42	42.11
LP3-D (II)	145.23	1628.76	90.09	74.01
LP5	73.72	268.87	340.49	92.34
LP7	6.71	20.92	144.76	30.71
LP8 (I)	330.46	327.42	627.20	270.42
LP8 (II)	236.44	206.93	132.30	77.05
LP8 (III)	50.32	27.95	109.21	19.57
AVG	135.93	332.34	238.74	83.30

TABLE B.9: Position errors (m) based on all combinations of five recordings with highest resolution for each roar and all combinations of references. Improvement is marked green and decline is marked red with respect to Table 5.7.

Method	Mean $\pm$ SD	Median	Max	Min
HD_PNT	38.26 ± 92.26	19.43	857.72	0.10
EMD	40.77 ± 91.13	20.96	857.72	0.10
CR_CRR (env)	40.38 ± 90.82	20.66	857.72	0.10
CR_CRR (wf)	0.12 ± 0.12	0.090	0.78	0.0071
GDOP	3.06 ± 2.33	1.93	9.10	1.12

TABLE B.10: Position errors (m) of 100 positions sampled over entire grid in square setup.

Method	Mean ± SD	Median	Max	Min
HD_PNT	217.57 ± 244.38	133.34	1193.14	8.72
EMD	227.21 ± 265.78	138.01	1193.14	8.72
CR_CRR (env)	228.48 ± 254.27	142.33	1193.14	8.72
CR_CRR (wf)	2.00 ± 6.96	0.41	55.09	0.0054
GDOP	16.74 ± 73.61	4.72	719.33	1.44

TABLE B.11: Position errors (m) of 100 positions sampled over entire grid in LP2 setup.

Method	Mean ± SD	Median	Max	Min
HD_PNT	74.46 ± 128.57	38.55	1156.77	0.96
EMD	87.11 ± 159.05	37.82	1156.77	0.96
CR_CRR (env)	70.81 ± 133.95	34.40	1156.77	0.96
CR_CRR (wf)	0.35 ± 0.40	0.18	1.97	0.0073
GDOP	4.87 ± 5.21	3.53	41.08	0.75

TABLE B.12: Position errors (m) of 100 positions sampled over entire grid in LP8 setup.

Bibliography

[1] Pankaj K. Agarwal and Micha Sharir. "Davenport–Schinzel sequences and their geometric applications". In: *Handbook of Computational Geometry*. Elsevier, 2000, pp. 1–47.

[2] Pankaj K. Agarwal et al. "Computing the discrete Fréchet distance in subquadratic time". In: *SIAM Journal on Computing* 43.2 (2014), pp. 429–449.

[3] Helmut Alt, Oswin Aichholzer, and Günter Rote. "Matching shapes with a reference point". In: *Proceedings of the Tenth Annual Symposium on Computational Geometry*. 1994, pp. 85–92.

[4] Helmut Alt, Bernd Behrends, and Johannes Blömer. "Approximate matching of polygonal shapes". In: *Proceedings of the Seventh Annual Symposium on Computational Geometry*. 1991, pp. 186–193.

[5] Helmut Alt and Maike Buchin. "Semi-computability of the Fréchet distance between surfaces". In: *Proceedings of the 21st European Workshop on Computational Geometry*. 2005, pp. 45–48.

[6] Helmut Alt and Michael Godau. "Computing the Fréchet distance between two polygonal curves". In: *International Journal of Computational Geometry & Applications* 5.01n02 (1995), pp. 75–91.

[7] Helmut Alt and Leonidas J. Guibas. "Discrete geometric shapes: Matching, interpolation, and approximation". In: *Handbook of Computational Geometry*. Elsevier, 2000, pp. 121–153.

[8] Helmut Alt, Christian Knauer, and Carola Wenk. "Matching polygonal curves with respect to the Fréchet distance". In: *Proceedings of the 18th Annual Symposium on Theoretical Aspects of Computer Science*. Springer. 2001, pp. 63–74.

[9] Valentin Andrei, Horia Cucu, and Lucian Petrică. "Considerations on developing a chainsaw intrusion detection and localization system for preventing unauthorized logging". In: *Journal of Electrical and Electronic Engineering* 3.6 (2015), pp. 202–207.

[10] Rinat Ben Avraham, Haim Kaplan, and Micha Sharir. "A faster algorithm for the discrete Fréchet distance under translation". In: *arXiv preprint arXiv:1501.03724* (2015).

[11] Chanderjit Bajaj. "The algebraic degree of geometric optimization problems". In: *Discrete & Computational Geometry* 3 (1988), pp. 177–191.

[12] Stephen Bancroft. "An algebraic solution of the GPS equations". In: *IEEE Transactions on Aerospace and Electronic Systems* 1 (1985), pp. 56–59.

[13] Kyle Barron. *pydelatin*. https://github.com/kylebarron/pydelatin. Version 0.2.8. Python package. 2024.

[14] Jean-Daniel Boissonnat and Mariette Yvinec. *Algorithmic Geometry*. Cambridge University Press, 1998.

[15] Karl Bringmann. "Why walking the dog takes time: Fréchet distance has no strongly subquadratic algorithms unless SETH fails". In: *Proceedings of the 2014 IEEE 55th Annual Symposium on Foundations of Computer Science*. IEEE. 2014, pp. 661–670.

[16] Karl Bringmann, Marvin Künnemann, and André Nusser. "Discrete Fréchet distance under translation: Conditional hardness and an improved algorithm". In: *ACM Transactions on Algorithms (TALG)* 17.3 (2021), pp. 1–42.

[17] Karl Bringmann and André Nusser. "Translating Hausdorff is hard: fine-grained lower bounds for Hausdorff distance under translation". In: *arXiv preprint arXiv:2101.07696* (2021).

[18] Karl Bringmann et al. "Dynamic time warping under translation: Approximation guided by space-filling curves". In: *arXiv preprint arXiv:2203.07898* (2022).

[19] Karl Bringmann et al. "Fine-Grained Complexity of Earth Mover's Distance under Translation". In: *arXiv preprint arXiv:2403.04356* (2024).

[20] Sergio Cabello et al. "Matching point sets with respect to the Earth Mover's Distance". In: *Computational Geometry* 39.2 (2008), pp. 118–133.

[21] L. Paul Chew and Klara Kedem. "Improvements on geometric pattern matching problems". In: *Proceedings of the Third Scandinavian Workshop on Algorithm Theory*. Springer. 1992, pp. 318–325.

[22] Christopher W. Clark, Peter Marler, and Kim Beeman. "Quantitative analysis of animal vocal phonology: an application to swamp sparrow song". In: *Ethology* 76.2 (1987), pp. 101–115.

[23] Scott Cohen and Leonidas Guibas. "The earth mover's distance under transformation sets". In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Vol. 2. IEEE. 1999, pp. 1076–1083.

[24] Richard Cole. "Slowing down sorting networks to obtain faster sorting algorithms". In: *Journal of the ACM (JACM)* 34.1 (1987), pp. 200–208.

[25] Anne-Sophie Crunchant, Jason T. Isaacs, and Alex K. Piel. "Localizing wild chimpanzees with passive acoustics". In: *Ecology and Evolution* 12.5 (2022), e8902.

[26] Rogério Grassetto Teixeira de Cunha et al. "Production of loud and quiet calls in howler monkeys". In: *Howler Monkeys: Adaptive Radiation, Systematics, and Morphology* (2015), pp. 337–368.

[27] Kevin Darras et al. "Autonomous sound recording outperforms human observation for sampling birds: a systematic map and user guide". In: *Ecological Applications* 29.6 (2019), e01954.

[28] Kevin Darras et al. "Measuring sound detection spaces for acoustic animal sampling and monitoring". In: *Biological Conservation* 201 (2016), pp. 29–37.

[29] Spiros Denaxas and Maria Pikoula. *spiros/discrete_frechet: meerkat stable release*. Version meerkat. Aug. 2019. DOI: 10.5281/zenodo.3366385. URL: https://doi.org/10.5281/zenodo.3366385.

[30] Leandro A. Do Nascimento, Cristian Pérez-Granados, and Karen H. Beard. "Passive acoustic monitoring and automatic detection of diel patterns and acoustic structure of howler monkey roars". In: *Diversity* 13.11 (2021), p. 566.

[31] Thomas Eiter and Heikki Mannila. *Computing discrete Fréchet distance*. Tech. rep. CD-TR 94/64. Christian Doppler Laboratory for Expert Systems, 1994.

[32] David Eppstein et al. "Improved grid map layout by point set matching". In: *International Journal of Computational Geometry & Applications* 25.02 (2015), pp. 101–122.

[33] José Manuel Fresno et al. "Survey on the performance of source localization algorithms". In: *Sensors* 17.11 (2017), p. 2666.

[34] Michael Garland and Paul S. Heckbert. *Fast polygonal approximation of terrains and height fields*. Tech. rep. CMU-CS-95-181. Carnegie Mellon University, 1995.

[35] Panos Giannopoulos and Remco C. Veltkamp. "A pseudo-metric for weighted point sets". In: *Proceedings of the 7th European Conference on Computer Vision – Part III*. Springer. 2002, pp. 715–730.

[36] Michael Godau. "On the complexity of measuring the similarity between geometric objects in higher dimensions". PhD thesis. Freie Universität Berlin, 1999.

[37] Kristen Grauman and Trevor Darrell. "Fast contour matching using approximate earth mover's distance". In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 1. IEEE. 2004, pp. I–220–I–227.

[38] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

[39] Frederick S. Hillier and Gerald J. Lieberman. *Introduction to Mathematical Programming*. McGraw-Hill, 1990.

[40] Daniel P. Huttenlocher and Klara Kedem. "Computing the minimum Hausdorff distance for point sets under translation". In: *Proceedings of the Sixth Annual Symposium on Computational Geometry*. 1990, pp. 340–349.

[41] Daniel P. Huttenlocher, Klara Kedem, and Micha Sharir. "The upper envelope of Voronoi surfaces and its applications". In: *Proceedings of the Seventh Annual Symposium on Computational Geometry*. 1991, pp. 194–203.

[42] IUCN. *The IUCN Red List of Threatened Species. Version 2024-1.* Accessed on 09 October 2024. 2024. URL: https://www.iucnredlist.org.

[43] Gabriel Jekateryńczuk and Zbigniew Piotrowski. "A Survey of Sound Source Localization and Detection Methods and Their Applications". In: *Sensors* 24.1 (2023), p. 68.

[44] Minghui Jiang, Ying Xu, and Binhai Zhu. "Protein structure–structure alignment with discrete Fréchet distance". In: *Journal of Bioinformatics and Computational Biology* 6.01 (2008), pp. 51–64.

[45] Shoken Kaneko and Hannes Gamper. "A fast forest reverberator using single scattering cylinders". In: *2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP)*. IEEE. 2021, pp. 1–5.

[46] Shoken Kaneko and Hannes Gamper. "Large-scale simulation of bird localization systems in forests with distributed microphone arrays". In: *JASA Express Letters* 2.10 (2022).

[47] Leonid V. Kantorovich. "On the translocation of masses". In: *Journal of Mathematical Sciences* 133.4 (2006).

[48] Dawn M. Kitchen et al. “Function of loud calls in howler monkeys”. In: *Howler Monkeys: Adaptive Radiation, Systematics, and Morphology* (2015), pp. 369–399.

[49] Oliver Klein and Remco C. Veltkamp. “Approximation algorithms for computing the earth mover’s distance under transformations”. In: *Proceedings of the 16th International Conference on Algorithms and Computation*. Springer. 2005, pp. 1019–1028.

[50] Jens C. Koblitz. “Arrayvolution: using microphone arrays to study bats in the field”. In: *Canadian Journal of Zoology* 96.9 (2018), pp. 933–938.

[51] Harold W. Kuhn. “The Hungarian method for the assignment problem”. In: *Naval Research Logistics Quarterly* 2.1-2 (1955), pp. 83–97.

[52] Sam Lapp et al. “OpenSoundscape: an open-source bioacoustics analysis package for Python”. In: *Methods in Ecology and Evolution* 14.9 (2023), pp. 2321–2328.

[53] Paul G. McDonald, Anahita J. N. Kazem, and Jonathan Wright. “A critical analysis of ‘false-feeding’ behavior in a cooperatively breeding bird: disturbance effects, satiated nestlings or deception?” In: *Behavioral Ecology and Sociobiology* 61 (2007), pp. 1623–1635.

[54] Brian McFee et al. *librosa/librosa: 0.11.0*. Version 0.11.0. Mar. 2025. DOI: 10.5281/zenodo.15006942. URL: https://doi.org/10.5281/zenodo.15006942.

[55] Nimrod Megiddo. “Applying parallel computation algorithms in the design of serial algorithms”. In: *Journal of the ACM (JACM)* 30.4 (1983), pp. 852–865.

[56] Gaspard Monge. “Mémoire sur la théorie des déblais et des remblais”. In: *Mem. Math. Phys. Acad. Royale Sci.* (1781), pp. 666–704.

[57] Axel Mosig and Michael Clausen. “Approximately matching polygonal curves with respect to the Fréchet distance”. In: *Computational Geometry* 30.2 (2005), pp. 113–127.

[58] Amir Nayyeri and Hanzhong Xu. “On computing the Fréchet distance between surfaces”. In: *32nd International Symposium on Computational Geometry (SoCG 2016)*. Vol. 51. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 2016, 55:1–55:15.

[59] Amir Nayyeri and Hanzhong Xu. “On the decidability of the Fréchet distance between surfaces”. In: *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM. 2018, pp. 1109–1120.

[60] Fabian Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[61] Bioacoustics Research Program. *Raven Pro: Interactive Sound Analysis Software (Version 1.5)*. 2014.

[62] Tessa A. Rhinehart et al. “Acoustic localization of terrestrial wildlife: Current practices and future opportunities”. In: *Ecology and Evolution* 10.13 (2020), pp. 6794–6818.

[63] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. “The earth mover’s distance as a metric for image retrieval”. In: *International Journal of Computer Vision* 40 (2000), pp. 99–121.

[64] John L. Spiesberger. “Hyperbolic location errors due to insufficient numbers of receivers”. In: *The Journal of the Acoustical Society of America* 109.6 (2001), pp. 3076–3079.

[65] John L. Spiesberger and Kurt M. Fristrup. "Passive localization of calling animals and sensing of their acoustic environment using acoustic tomography". In: *The American Naturalist* 135.1 (1990), pp. 107–153.

[66] Brigitte Spillmann et al. "Validation of an acoustic location system to monitor Bornean orangutan (Pongo pygmaeus wurmbii) long calls". In: *American Journal of Primatology* 77.7 (2015), pp. 767–776.

[67] Kathleen M. Stafford, Christopher G. Fox, and David S. Clark. "Long-range acoustic detection and localization of blue whale calls in the northeast Pacific Ocean". In: *The Journal of the Acoustical Society of America* 104.6 (1998), pp. 3616–3625.

[68] Dan Stowell et al. "Automatic acoustic detection of birds through deep learning: the first bird audio detection challenge". In: *Methods in Ecology and Evolution* 10.3 (2019), pp. 368–380.

[69] Abdel Aziz Taha and Allan Hanbury. "An efficient algorithm for calculating the exact Hausdorff distance". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.11 (2015), pp. 2153–2163.

[70] Pravin Vaidya. "Geometry helps in matching". In: *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*. 1988, pp. 422–425.

[71] Remco C. Veltkamp and Michiel Hagedoorn. "State of the art in shape matching". In: *Principles of Visual Information Retrieval* (2001), pp. 87–119.

[72] Suresh Venkatasubramanian. "Geometric shape matching and drug design". PhD thesis. Stanford University, 1999.

[73] Régine Vercauteren Drubbel and Jean-Pierre Gautier. "On the occurrence of nocturnal and diurnal loud calls, differing in structure and duration, in red howlers (Alouatta seniculus) of French Guyana". In: *Folia Primatologica* 60.4 (1993), pp. 195–209.

[74] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[75] Siebe Vuijst. *Matching under One-Dimensional Translations*. https://github.com/SVuijst/ShapeMatching. Python package. 2025.

[76] Carola Wenk. "Shape matching in higher dimensions". PhD thesis. Freie Universität Berlin, 2003.

[77] Matthew Wijers et al. "CARACAL: A versatile passive acoustic monitoring tool for wildlife research and conservation". In: *Bioacoustics* 30.1 (2021), pp. 41–57.

[78] David R. Wilson et al. "Sound Finder: a new software approach for localizing animals recorded with a microphone array". In: *Bioacoustics* 23.2 (2014), pp. 99–112.

[79] Rory P. Wilson and Clive R. McMahon. "Measuring devices on wild animals: what constitutes acceptable practice?" In: *Frontiers in Ecology and the Environment* 4.3 (2006), pp. 147–154.