

# Gunshot Detection & Localization for Anti-Poaching Initiatives

Master of Arts Exegesis - April 2019

Kyle Hoefer

Arizona State University Digital Culture - Arts, Media, and Engineering Herberger Institute for Design and the Arts

**Committee Chair** 

• Garth Paine, Associate Professor - School of Arts, Media, and Engineering



## **Table of Contents**

	Introduction	3
1.	Background	4
	1.1 Location background & history	4
	1.2 Design considerations, challenges, and needs	5
	1.3 The acoustics of ballistics	6
	1.4 Previous detection systems	7
	1.5 Low cost, low power, low data	8
2.	Spectral Detection Parameters	9
	2.1 Frequency analysis of a gunshot	9
	2.2 Amplitude & loudness monitoring	13
	2.3 Adaptive background subtraction	15
	2.4 Importance of Spectral Centroid	15
	2.5 The vector of change	17
3.	Preliminary Testing	20
	3.1 First controlled data acquisition	20
	3.2 The desert versus the forest	20
4.	On-Site Recording and Analysis	22
	4.1 The recording process	22
	4.2 Regional discoveries	23
	4.3 The inverse effect of energy	24
	4.4 Validation of the vector of change	27



5.	Building Code	28
	5.1 Why Teensy 3.6	28
	5.2 Initial MATLAB algorithm principles	29
	5.3 Calculating the FFT & energy	29
	5.4 Calculating spectral centroid	30
	5.5 Vector math in C/C++	32
	5.6 Parameter verification & shot detection	34
6.	Final Testing & Results	35
	6.1 Accuracy of detection	35
7.	Future Considerations	37
8.	Conclusion & Acknowledgments	39
	Reference	40
	List of Figures	42
	Appendix	43
	Appendix A: Literature Review	43
	Appendix B: Comparison of FFT window types	50
	Appendix C: Spectral feature extraction code in MATLAB	51
	Appendix D: Final gunshot detection algorithm	52



### Introduction

The following paper details the research and implementation of a gunshot detection algorithm for an on-going anti-poaching project in Costa Rica, launched by the Acoustic Ecology Lab<sup>1</sup> at Arizona State University in conjunction with conservation researchers at The Phoenix Zoo<sup>2</sup>. This project involves solar powered microphone units and wireless transmission of predicted gunshot locations through a proposed mesh network, to track illegal poaching as it occurs in the private & protected land of Las Alturas Del Bosque Verde. The text outlines from beginning to end; acoustics research completed in the realm of ballistics, analysis of spectral parameters and their use in previous gunshot detection endeavors, proposed novel combinations of these parameters for accurate long distance detection, on-site field recordings and analysis, the building of code to utilize microcontroller development boards as a means of real-time detection monitoring, and tested results on the reliability and accuracy of this code. Future considerations are also included to easily implement these algorithms with the parallel research stream of localization. Although a site-specific application, the algorithm proposed in this text aims to create a robust set of variables which can be applied to any sonic environment, coupled with a low cost, low power, low data microphone-based monitoring unit.

<sup>&</sup>lt;sup>1</sup> The Acoustic Ecology Lab @ ASU. (n.d.). Retrieved on April 12, 2019, from https://acousticecologylab.org/

<sup>&</sup>lt;sup>2</sup> The Phoenix Zoo. (n.d.). Retrieved on April 12, 2019, from https://www.phoenixzoo.org/



### 1. Background

#### **1.1 Location Background and History**

Las Alturas Del Bosque Verde is a privately owned, ten-thousand hectare (24, 171 acres) plot of land turned animal sanctuary in the Puntarenas region of Southern Costa Rica, bordering the country of Panama. It is host to many research stations and worldwide conservation endeavors from The Phoenix Zoo to Spain's ProCAT<sup>3</sup> (Proyecto de Conservación de Aguas y Tierras). This inland region of Costa Rica resides at approximately 4,330 feet (1,320 meters), and it's rather high elevation makes the area unique to other parts of the country. Although still considered rainforest, its dry season spans six months out of the year and is characterized by moderately comfortable humidity levels of around fifty percent. Primarily due to these humidity levels paired with a dense forest environment, it boasts a rich history in coffee farming and a large variety of animal species. Although its abundant levels of relatively rare species such as white-lipped peccary and jaguar are positives, the region has also been subject to poaching.

As a private organization, Las Alturas employs local workers as security guards to protect against intruders attempting to poach wildlife and interfere with coffee farming. However, due to the sheer size of this sanctuary and the fact that many public offroads intersect the private land, it is nearly impossible to catch these poachers in the act. There are simply too many roads and insufficient personnel to safely guard all the highly poached areas. An added level of concern, the local village is small enough so that poachers learn the movements and schedules of the guards on duty. This allows the intruders to not only avoid them while on the preserve, but also the guards and their families targets in town. It is not uncommon to hear from workers of run-ins with these intruders that contain instances of being shot at and harassed, on and off the private land.

Because of this concern, efforts are being made to autonomously monitor the region for species and hunters through motion-only based camera traps installed on the base of trees. While somewhat helpful, various issues have arisen - cameras must be fitted with large data SD cards, and the pictures written to these cards can only be viewed on a computer when the camera has been physically accessed and cards collected. The camera's line of sight is extremely limited resulting in over one-hundred cameras needing to be placed and serviced. It can only capture movement in a short period of time meaning a picture of poachers passing by from three weeks ago does not give them info as to where the poaching occurred. Lastly, these

<sup>&</sup>lt;sup>3</sup> PROCAT - PROYECTO DE CONSERVACIÓN DE AGUAS Y TIERRAS. (n.d.). Retrieved on April 7, 2019, from http://procat-conservation.org/



camera units are not cheap and poachers are able to spot and destroy them due to their low-lying placement on the trees, even when encased in steel boxes built by the workers. After the Acoustic Ecology Lab at ASU was approached by head conservationists at the Phoenix Zoo and discussing the weaknesses of current surveillance methods, this research examining the possibility of adding sound and spectral analysis of gunshots to existing methods was initiated. A successfully built system would allow the security detail to gather information of poaching remotely and safely in real-time, and be alerted to the location of gunshots all without tedious trips to service cameras or listening devices.

The methods which were chosen for testing were largely garnered from listening methods practiced in acoustic ecology. The field of Acoustic Ecology was defined by R. Murray Schafer in the 1960's and focuses on the relationship of humans and their environment through sound. The Acoustic Ecology Lab at ASU is an initiative to bring awareness to listening through project-based applications with a large emphasis on community outreach and engagement [30]. The co-founder of this lab, Dr. Garth Paine details the importance of these aspects through his publications such as, "Listening to nature: How sound can help us understand environmental change," in which he outlines ways listening could benefit current conservation research. Current monitoring methods have large reliance on sight. "Other factors, such as changes in a forest's foliage density from spring to fall, also change a site's reverberation characteristics. Exploring these qualities has led me to think about how perceptual measures of sound inform our understanding of environmental health, opening a new angle of inquiry around psychoacoustic properties of environmental sound" [31]. The psychoacoustic properties of environmental sound, as are stated by Paine, the leading reason for the specific methodology taken in this project. Rather than listen to match incoming signals to predetermined templates or masks for specific sonic cues, the research is informed by this concept of deeper listening, which revolves around taking in and hearing the environment as a whole, learning to use the sonic features which already exist within it to an advantage.

#### **1.2** Design considerations, challenges, and needs

While the application of this project is very specialized to reflect a certain region and precise variables associated with this region, it has always been imperative that the means by which this system is created is one that can be applied broadly to other regions plagued by similar issues. This creates a "non-hardcoded" system which can be applied anywhere, and is not limited to a one-time use scenario.

**Upkeep**: It is difficult to travel across the sanctuary's terrain. It was clear from the beginning of this project that any system must be self sustaining for an extended period of time without service. The need to consistently service any surveillance unit in this area would make it less useful than not having one at all, as time and effort would be taken away from patrolling and be exhausted on upkeep. A potential solution to this problem was the use of solar to



charge and maintain battery power, discussed in section 1.5.

**Location:** The placement of existing cameras led to them being destroyed. Their placement required line of sight to the object they are trying to capture. This issue can be mitigated through the application of audio, as a microphone does not need to be directly in view of whatever it is capturing, so long as its surroundings do not obstruct the sound from reaching it. Because of this, it was decided that the system must be installed out of sight, but not obstructed, high along the treeline canopy of the forest. This location also allows for easier installation of a solar unit, as sun rarely passed through the dense rainforest canopy.

**Weather:** Although the vast majority of poaching is throughout the six-month dry season, there are still instances where rain and high humidity levels could affect performance and accuracy of the proposed system. Proper protection of the microphone, microprocessor, solar charging station, radio communications, and battery is required to keep moisture out but still allow necessary audio frequencies to pass.

**Scale:** It was clear from the beginning that due to the size of this plot of land, it would be nearly impossible to cover all of it. The previous camera surveillance has proven high traffic areas for poaching due to the public offroads, and there are a few sections of specialized plots (reaching an extent of approximately 20-25 kilometers), which poachers tend to gravitate to.

**Noise:** The Costa Rican rainforest is home to an extensive range of creatures, some being extremely loud. Because this forest is not a quiet place, we realized that sonic occurrences extremely close to the microphone (howler monkeys, rain, crickets, rushing rivers, wind, etc.) could compromise and overpower any gunshot sound which occurred many kilometers away. Because of this, extra consideration would need to be made in the detection algorithm to distinguish background sound from sonic events of interest.

#### 1.3 The acoustics of ballistics

The root of this project lies in the sonic makeup of a gunshot. Because this land is so sonically untouched by man, it was important to first learn what characterizes a gunshot and how it will travel across the many miles of this specific landscape. All information presented in subsections 1.3 and 1.4 can be found more in-depth in this projects literature review in Appendix A [28]. As stated in the review, it is clear through the work of Robert Maher [1][2], that firearms present three sonic events upon being discharged. These include the mechanical action, muzzle blast, and bullet shockwave. The mechanical action references the cocking mechanism on various semi-automatic rifles. In this project's case, previous evidence has proven poachers use bolt-action rifles as they are cheaper to purchase and provide more accuracy for hunting game. Bolt-action rifles fire a single shot and require manual cocking and



reloading, therefore the semi-automatic mechanical action event has been ruled out. The muzzle blast occurs as the explosion of gunpowder propels the bullet out of the chamber. This event lasts around three to five milliseconds and is always louder when facing the barrel of the gun, although the energy wave is dispersed spherically at the speed of sound. Bullet shockwaves are created when the bullet reaches or surpasses the speed of sound. These waves typically last two-hundred microseconds and propagate outwards from the bullet's path at its highest speed, becoming increasingly parallel to the bullet as it begins to slow [3]. Although amplitude variation will occur depending on the direction of the shot, shockwaves will always reach a specific location prior to the muzzle blast if the bullet surpasses the speed of sound.

It is well known from the confiscation of weapons from the poachers that the caliber of choice when hunting small game such as the peccary is the .22 long rifle. While hunting larger game such as the jaguar, a larger caliber ranging from 9mm to the more easily accessible .223 or .308 has been found. However, the tradeoff with these larger, faster, rifle calibers is that it can maim the animal unintentionally depending on the bullet's path, destroying the coat or pieces of the animal which are important to the poachers. There is a specific set of .22 caliber ammunition titled sub-sonics that operate below the speed of sound (approximately 1,125 feet per second), these are much quieter as they avoid the supersonic bullet crack. This round would significantly decrease the sound made by the poachers, but the low bullet travel speed paired with smaller round would not necessarily guarantee a kill on even small game due to its smaller energy transfer upon impact. Because of this, it was ruled out of being a concern.

Upon first describing a gunshot, one may say that it's loud and "boomy" at a significantly close distance. Further away it might be quieter, but one may still say they feel that boom in their chest, and this is what makes humans good at distinguishing a gunshot from any other loud sound. It was made clear through ballistics research that the key to creating a footprint of a gunshot is in it's "rise time." That is the 200-microsecond window following the muzzle blast where the bullet breaks the speed of sound. Such a quick rise and fall of energy emitted by this event is something which never occurs in nature, and is a key variable which distinguishes a shot from all other sound sources in the rainforest.

#### **1.4 Previous detection systems**

Extensive research of prior gunshot detection systems has proven that none of them truly capitalized on the very fast energy profile outlined above. The one consistency across the majority of existing systems is a host of complicated power-consuming algorithms. These detection algorithms could range from the Mel Frequency Cepstral Coefficients to adaptive background noise cancellation through multiple layers of notch & bandpass filtering [6]. Unlike the detection, the triangulation (location) through TDOA<sup>4</sup> (Time-Difference on Arrival) of the gunshot must be calculated through consistent speed of sound calculations and generalized

<sup>&</sup>lt;sup>4</sup> Shaw, G. S. (n.d.). Multilateration (MLAT). Retrieved on April 5, 2019, from http://www.multilateration.com/surveillance/multilateration.html



cross-correlation phase transforms. While these are computationally intensive tasks, this aspect may be handled by a computer receiving the data, and not on the processors in the field, allowing them to be purely used for detection. A good example of an expensive, large data, non-autonomous gunshot detection system currently on the market is ShotSpotter<sup>5</sup>. This product is an urban-based gunshot detection system meant to capture and alert gunfire to accompanying authorities through the placement of microphone systems on multiple buildings throughout the city. While their triangulation methods are similar to those being proposed here, there are many pitfalls. Firstly, Shotspotters microphone devices are always recording, this is because the ultimate decision of whether or not a gunshot was produced is made by a certified "acoustic expert" who is standing by as a dispatcher listening to any and all detections that come through. The fact that these devices are discreet, hidden, and always recording has raised concerns regarding non-authorized breaches of public privacy. Also, the use of a human dispatcher removes the autonomous part of this system, and while the detection and location algorithms may help speed up the process of localization of the source, this proves that the model is not robust enough to provide certainty of a gunshot versus other sounds without some sort of human decision making. Lastly, the average cost of ShotSpotter is approximately \$65,000 - \$95,000 per square mile per year. This means that currently, with the city of Oakland's 16 miles covered by ShotSpotter, they are paying an approximate minimum of 1.04 million dollars a year to keep this system up and running.

#### 1.5 Low cost, low power, low data

Although 2.4 million dollars a year to pay a dispatcher to report potential gunshot locations in the entirety of Las Alturas Del Bosque Verde sounded tempting, this wouldn't be a viable option.

With a majority of previous systems revolving around the same technical design of ShotSpotter built nearly twenty years ago, it became clear that a number of innovations needed to be made to fit the cost and reliability requirements outlined in Las Alturas:

**Low cost:** As an independently funded project with minimal help through the initial design and prototyping stages, everything must be kept as low cost as possible. This does not only apply to materials but also operation. Fixing regular issues can become a costly endeavor, so building something reliable and heavy duty is key.

**Low power:** As stated above and in section 1.2, reliability is a must and this goes hand in hand with efficient and minimal power consumption. Although solar charging for battery maintenance is a possibility, the fifty percent sunlight that this region receives per year does not guarantee that it will be sufficient to keep these devices functioning throughout every night. Every step of this project through building and coding must address low power consumption as a priority.

<sup>&</sup>lt;sup>5</sup> ShotSpotter. (n.d.). Retrieved on January 7, 2019, from https://www.shotspotter.com/



Low data: With the two previous caveats taken into account, it was made clear that the final listening units which will be dispatched to locations high in the forest canopy will not be able to record and transmit audio files twenty-four hours a day over long distances through wireless communication. Steps needed to be made to take this incoming audio data in short amounts, run the calculations to verify shots quickly, and when a positive detection is returned, only the numerical values associated with the variables measured should be transmitted with the unit ID, then store the audio of the detected shot on a micro-SD card and move on to the next frame.

The three sections above provide reasoning as to the difference between the proposed system documented in this exegesis and to the ShotSpotter system. Lack of human verification means that the detection algorithms must be extremely robust and rule out all false positives or missed detections. Reliability must be at the highest possible level as all unit's dispatched would be in remote areas of the rainforest not easily reachable, and cost must be kept low so that multiple listening devices can be placed accordingly to cover the required area of high traffic poaching.



### 2. Spectral Detection Parameters

#### 2.1 Frequency analysis of a gunshot

As stated in chapter one, the root of this project relies upon the sonic makeup of a gunshot. This analysis relies on several key DSP feature extraction techniques. Before delving into these extractions, it is important to look at the base algorithm, the Fast Fourier Transform, or "FFT" for short.

**FFT:** The Fast Fourier transform is a class of algorithm based around the computational optimization of the discrete Fourier transform (DFT), which is a group of equations allowing us to transform any signal which resides in the time domain (on this occasion gunshot recordings), to the frequency domain [17]. There are a few key parameters that must be taken into consideration when performing this function. These include sampling rate, Nyquist frequency, window size, window overlap, window enveloped, FFT size, and bin size.

Sampling Rate: The sampling rate defines the average number of audio samples per second, this is specifically referenced in Hertz (Hz). The larger number of samples per second, the larger range of frequencies captured. As an example, telephone communication is limited to 8,000Hz to preserve data size. Most CD quality audio has a sampling rate of 44.1kHz, while DVD and Blu-ray audio can have rates of 96kHz, or even up to 196kHz [18].

Nyquist Frequency: The reason for these very specific sampling rates is in part due to the Nyquist theorem. This theorem states that in order to properly convert audio in an analog-to-digital conversion (ADC), and then reproduce the same signal using digital-to-analog converter(DAC), the sampling rate must be two times the highest frequency desired [19]. If this value is not met, it can introduce aliasing and therefore unwanted distortion into the signal. The average range of human hearing spans from 20Hz to 20,000Hz, meaning the lowest sampling rate required to produce all frequencies humans can hear is 40kHz. Any sampling rates past this value contain ultrasonic frequencies which cannot be heard by humans. In order to gather the largest possible amount of insight on the frequencies exhibited by the gunshot in initial testing, a sampling rate of 96kHz was chosen, giving a frequency range up to 48kHz, well into the ultrasonic range



<u>Windowing</u>: When splitting a signal with non-periodic data from the time domain to the frequency domain, unwanted instances of spectral leakage can occur. This leakage can cause the signal to be redistributed over the entire frequency range, muddying the analysis of the



Figure 2.1: Frequency spectrum of sine wave aligning with frequency resolution (red) and sine wave not aligning with frequency resolution (green) [19]

amplitude of the desired range [18]. This loss in amplitude due to spectral leakage can be viewed in Figure 2.1. By applying a windowing function, this forces a smoothing of the data at the start and end of the progression, allowing for a more accurate analysis of amplitude. There are various windowing types which can be applied, for a full graph of examples detailing each window type, see Appendix B. In order for windowing to be applied appropriately, the window length must match the FFT size. For the purposes of this project, the Hann window type was chosen, with a length of 1024 samples.

<u>FFT & Bin Size</u>: Before the FFT can be computed, it must collect a certain number of samples to be analyzed - this is known as the FFT size, or length. Common values of FFT length range from 1024, 2048, 8192, and even 16,384. The bin size references the number of bins, or the collections of frequencies that the FFT will be split in to. The bin size varies as a function of the sampling rate and respective Nyquist frequency, and FFT size, and can be calculated as follows:

$$\frac{Nyquist\ Frequency}{FFT\ Frame\ Size} = Bin\ Size$$

**However, there is a catch** - the longer the FFT length the higher the resolution of the frequency analysis, but the longer time it will take to compute. If attempts to analyze a quick sound are being made, a shorter FFT length will give better temporal resolution, but the bin size (frequency resolution) will be larger and less accurate. If a longer FFT length is used then a smaller (more accurate) bin size is produced, but the event analysis could be skewed due to unwanted sonic events which occur after the primary sound event. This tradeoff is a great concern for this project, as it was made clear from the previous acoustics research that gunshots are extremely quick sonic events happening in under a fifth of a second. However, as much of the initial energy in the gunshot resides at low frequencies, a high frequency resolution is required at low frequencies. A large FFT window size is required in order to produce this resolution, which works against the temporal resolution. Because there is no perfect solution to



this problem, an FFT length and bin size must be computed which favors low computational power, but enough resolution to distinguish the lower frequency energy.

To begin with testing, a recording of a random gunshot at an unknown distance was recorded at 96kHz sampling rate at a local shooting range. This audio was processed using MATLAB, and two FFT sizes were chosen to compare their ability to distinguish critical frequency bands.



Fig 2.3 FFT of gunshot at length 16,384



#### FFT Length (Samples)

#### <u>1024</u>

<u>16,384</u>

 $\frac{48,000Hz}{1024} = 46.88Hz \ per \ Bin$ 

 $\frac{48,000Hz}{16,384} = 2.93Hz \ per \ Bin$ 

The graphs and tables above display stark differences in analysis for each length choice. In Figure 2.2 there is a visibly lower resolution line, however, due to the quick sample collection, the low frequencies are much more prevalent and nearly twelve times as large at 40Hz in relation to 500Hz. The table also displays that the resolution of Hertz per bin is nearly 47. This is not ideal as it means that from 0Hz to about 3000Hz (where the gunshot analysis is most critical), there are only about 63 values of averaged amplitude. If a comparison of this data is made with Figure 2.3, the graph is much more detailed, but there is a large spike in the 400Hz to 700Hz range that is even louder than the subsonic values of about 40Hz that are of greater interest. This spike could be due to the long sample collection period picking up sonic events that aren't gunshots, clouding the analysis. One upside to this calculation is the width of each analysis, sitting at about 3Hz. With this resolution, there are approximately 1,023 values of averaged amplitude from the range of 0Hz to 3,000Hz.

With all these variables taken into account, an FFT length of 1024 samples was chosen for this project with a window overlap value of twenty-five percent. The first bit of reasoning for this stemmed from the original concept of low data and low power. The computational power to perform the larger length calculation is nearly sixteen times that of its smaller counterpart. Secondly, the quick rise and fall of the gunshot is the most crucial piece of information, and by extending the window size, temporal smearing would make the analysis unreliable as the readout would be muddy and include sounds that we are not interested in analyzing. All this considered, it is much more beneficial in this instance to focus on the quick sampling period over frequency resolution.

#### 2.2 Amplitude and loudness monitoring

Following the FFT calculation, spectral feature extraction parameters were chosen to discern a gunshot from naturally occurring sounds, the first of these being amplitude (also known as energy). On its own, the amplitude is the difference between the highest and lowest points of a signal in comparison to its equilibrium, often described in units of Decibels. In regards to the way humans perceive sound, the larger the amplitude, the louder the sound.



One of the easiest ways to analyze some initial examples of amplitude was through the program Sonic Visualizer<sup>6</sup>, utilizing Jamie Bullock's lightweight feature extraction library LibXtract<sup>7</sup> [20]. The extraction used within this toolbox was named "Loudness." Although amplitude and loudness are not the same, they are related. While amplitude is a value which can be precisely measured and recreated, loudness is a perceived psycho-acoustic measurement and not perfectly definable. This feature takes into account multiple other factors such as sound pressure level and time-behavior of the sound, meaning that a sound will not be exactly the same loudness level for all individuals [21]. With this being said, loudness



next.

Fig 2.4 Loudness analysis in Sonic Visualizer, over a period of three gunshots was still a viable means to analyze the random gunshot recording collected to gather an idea of what the variance in energy looked like when the shot was taken. The green line on the adjacent graph displays the loudness value over a period of several shots. This is the same recording used in the FFT example in 2.1, however, it includes all three of the shots captured and not just the initial one. There is a visible difference displayed each time the shots shockwave hits the microphone, causing a loudness spike which is approximately twice as loud from one frame to the

There are several factors that contribute to the successful analysis in this instance which will not always carry over to other recordings. Firstly, the loudness level of the surrounding environment is very

low when the shot occurs, causing a more noticeable spike. This spike will be much smaller if the gunshot occurs further away, and can easily be masked out by any sound which is closer to the microphone. Even if this unwanted sonic event is identifiably softer than the shot, it will be perceived as louder due to its proximity. Secondly, the algorithm used to calculate loudness in this instance takes the full audio spectrum into account. It was made clear from the FFT that much of the energy in a gunshot is subsonic, and any energy recorded above these desirable frequencies will continuously provide false readings and incorrectly vary the feedback.

<sup>&</sup>lt;sup>6</sup> Sonic Visualiser. (n.d.). Retrieved January 16, 2018, from https://www.sonicvisualiser.org/

<sup>&</sup>lt;sup>7</sup> Bullock, J. (2008). *Implementing audio feature extraction in live electronic music*.. Birmingham City University. Retrieved March 20, 2019, from https://www.academia.edu/4493811/Implementing\_audio\_feature\_extraction\_in\_live\_electronic\_music



The issue of needing to only focus on the analysis of the lower part of the spectrum has a relatively simple fix in theory, as filtering can be used to only pass through the analysis on the frequencies we desire. As an example, a low-pass filter will only allow analysis to be made on and below the frequency 1500Hz. This effectively rules out sounds such as high-pitched bird chirps, insects, or unwanted electrical noise. There is still a host of sounds which could be seen as a problem; cars, planes, wind, and other animals all contain energy in the 0Hz to 1500Hz range. For these reasons loudness on its own is not a viable means of detection, but provides a piece of information that can fit into a larger puzzle.

#### 2.3 Adaptive background subtraction

There is a possibility of introducing background subtraction to remove unwanted constant frequencies on an ever-changing, always adapting basis. By taking spectral snapshots, or averages over periods of time to analyze constant frequencies in the spectrum that are undesired, notch filters can be applied to cut out these instances. A positive impact from this could be completely removing the harmonics of the river rushing through the preserve from the analysis. While this is useful, it will still only aide in constant sounds over long periods of time, issues like animal calls, wind, and passing trucks will still bypass this protection.

#### 2.4 Importance of spectral centroid

While extraneous and unwanted higher frequency sounds may be an issue for monitoring loudness, there are some extractions that take advantage of this energy, the most important one being the spectral centroid. This algorithm allows the calculation of the "center of mass" of the frequency spectrum through values which were previously decoded through the FFT. While the FFT reports energy levels in each of the bins that have been created (512 in this case), one can find the spectral centroid for that frequency snapshot by multiplying all the bin's center frequencies (ex. Bin 1 = 43HZ, or (0 - 43), meaning its center would be 21.5Hz) to their total energy values, then dividing by the sum of their energy values. This is displayed below:

Centroid = 
$$\frac{\sum_{n=0}^{N-1} f(n) x(n)}{\sum_{n=0}^{N-1} x(n)}$$

In this instance, x(n) represents the weighted frequency value, or magnitude, of bin number n, and f(n) represents the center frequency of that bin [22].



What this equation spits out is a value in Hertz that represents the average center of mass for that period of time, dependent on FFT size. Different environments have varying spectral centroid values over time. For example, a busy highway might have a very low spectral centroid during rush hour times due to the rumbling of car tires on the road, but at night as

fewer cars travel the spectral centroid will increase and rest somewhere more equivalent to the natural sounds around it. Because of this, if a low-pass filter or adaptive set of notch filters are applied to the incoming sound, the spectral centroid will be incorrectly weighted, and small changes might not be as prevalent. This sparked an interest as previous surface level research proved that a majority of the creatures occupying the sonic space of the rainforest landscape are insects which tend to emit higher frequencies. During periods of sudden subsonic energy, a clear drop in the Hz value of spectral centroid should



Fig 2.5 Spectral centroid analysis in Sonic Visualizer, over a period of three gunshots

occur. Performing this initial analysis using the LibXtract toolkit provided a bit of a lackluster result on the same audio used to detect loudness, as observed in Figure 2.4. The centroid seems to hover back and forth between ~1300Hz and ~3400Hz. The change is hardly noticeable on its own, so much so that it is impossible to distinguish where exactly the shots occur without including the waveform of the audio file. This is partially due to the location of the microphone being inside a vehicle and having close to no gain and picking up no background noise, leaving the average hovering value of the spectral centroid to be very low to begin with.

However, this becomes more distinguishable if the graph of loudness is included on top as shown in Figure 2.5. Due to these purple loudness spikes, it is observable where there are inverse correlations in spectral centroid drops. It's clear that every time the loudness increases, there is a decline in the centroid. Even though both the centroid and loudness are still a bit random on their own, when working together they provide a more reliable and appropriately detectable graph within Sonic Visualizer.





three gunshots

#### 2.5 The vector of change

Arguably the most important piece of analysis to this detection puzzle is the vector of change. Previously explained sets of feature extraction would rely only on thresholding. This means that once the loudness or spectral centroid values pass a desired threshold (either separately or in unison), a shot will be detected. While this is useful for test cases, it is a very trivial and hard-coded method that will not adapt well to change, and only takes into account a binary means of detection. This method relies on the current frame and does not look at any frames which occurred before it when computing this threshold. Because of these graphs, it is beneficial to view a simple set of numbers on a time-based scale. Remove Sonic Visualizer, and what is observable through the graphs versus what the computer sees are different entities. By simply looking for a target threshold to be passed, the quick rise and fall time of the gunshot has been disregarded. A viable way to convert what is seen in these graphs to be understood by the computer is by examining the rate of change as a vector.



When breaking down the graphs created by Sonic Visualizer, it is observable that there are two dimensions associated. The X dimension is a constant value at which a calculation occurs, (technically this is the FFT length), while the Y dimension is the value reported for that frame and always changing.

*Magnitude:* The graphs display lines from frame to frame, and these lines are known as the magnitude. For the magnitude to be calculated, it is required to have a comparison of the previous frame to the current frame. As an example, calculating the magnitude of vectors' A to B can be written as:

$$\left| \xrightarrow{AB} \right| = \sqrt{\left( x_2 - x_1 \right)^2 + \left( y_2 - y_1 \right)^2}$$

In the case of loudness, two example frames A = (5, 2.1) and B = (10, 7.8) would look like

$$\left| \xrightarrow{AB} \right| = \sqrt{(10-5)^2 + (7.8 - 2.1)^2}$$
$$= \sqrt{5^2 + 5.7^2}$$
$$\approx 57.49$$

Because the X value will always be a constant, all that is occurring to find the magnitude is subtracting the current Y value from the previous. Because the magnitude is only reporting the distance of the line, the value will always be positive.

**Direction:** The other output of the vector of change algorithm is the direction. While the magnitude is the length of the line, the direction is the angle of the line from the previous frame to the current, in reference to a horizontal line which is equal to the previous frame. The rules state that if this angle is larger, up to 90 degrees, the larger the magnitude and therefore steeper the change. The direction of the vector can be found by calculating:

$$\tan\theta = \frac{y_2 - y_1}{x_2 - x_1}$$

For the same frames listed for magnitude, this would equate to

$$\theta = \tan^{-1} \left( \frac{5.7}{5} \right)$$
$$\approx 49^{\circ}$$



Unlike the magnitude, the directional vector calculation can report negative directions in degrees. Because of this, an extra layer of detection is added as it is only required to look for steep positive variation in loudness in conjunction with steep negative variation in spectral centroid. If there is a steep negative direction change in loudness, and a positive change in centroid, it can be ignored. With the addition of these vector calculations along with the thresholding values, a dense layer of detection has been created which relies on over six variables of criteria to be met before a gunshot is reported. However, before being able to test this theory, collections of recordings must be made to assure that the loudness and spectral centroid measurements will hold true in Sonic Visualizer over a tested data set. It is crucial to verify whether these extractions will hold true, and observe just how well they will consistently perform over a large variety of distances from the shooter.



## 3. Preliminary Testing

#### 3.1 First controlled data acquisition

Up until this point, these various spectral parameters have been tested using a single recording of an unknown caliber firearm from an unknown distance. In order to properly begin tests, baseline recordings must be made in a prepared environment, so that distances and changes of the bullets sound over the environment can be noted accordingly. These recordings took place in the recreational shooting region of the Four Peaks park in Arizona during the latter portion of winter. Test shots were performed on a .223 caliber bolt-action rifle, recorded using an iPhone for ease with the sampling rate of 96kHz. Due to the inability of remote recording using the devices at hand, tests were performed from the distances of 30m to 500m away from the shooter.

#### 3.2 The desert versus the forest

These baseline tests were useful because they proved that the spectral parameters chosen held true over multiple instances of gunshots at a variety of distances. What these tests are lacking is the impulse response of the rainforest environment. An impulse response is loosely defined as the way a certain environment affects the way you hear the sounds within it. This response can be varied by the physical makeup of the space, as well as certain parameters such as temperature, humidity, etc. When comparing the impulse responses of the desert environment and the rainforest, they are nearly opposites.



The region in Arizona where initial gunshots were recorded included a vast number of rolling hills around one-hundred to two-hundred feet high, covered in large rocks with not much more than low-lying dried out shrubs. Access to this area was confined to the valleys of the rolling hills due to the recreational shooting rules. Because of safety protocol, the public must only shoot into the sides of these hills so that the possibility of stray bullets will not harm others. The rocky, tree-less makeup of these areas brought with it a very reflective atmosphere, and sounds bounced off the sides of the hills with ease. Because of this, sound waves were carried far and reverberated for many seconds.

Fig 3.1 The initial testing location in Four Peaks, Arizona



It is not uncommon to feel like a gunshot which occurs up to a mile away is occurring within half that distance or less. Since the psycho-acoustic measurement of loudness was being analyzed to begin with, this phenomenon was verified by multiple people who have stood in these valleys, many scared that the gunshot was too close to them for comfort. To make matters worse, the desert climate in the middle of a winter day is still extremely dry and relatively warm. Sound moves through hot air much faster than cold because it is less dense, but dry air absorbs much more energy than humid air, making it weaker [23]. Even though this occurrence only affects frequencies at and above 2,000Hz, it could have a significant effect on how far the higher frequency bullet crack carries at longer ranges.



Fig 3.2 Shots were recorded with a .223 bolt-action rifle

Although information has been obtained to recreate the exact distances at which the poachers' gunshots occur, the energy readings and frequency responses will be varied in the rainforest. This forest landscape is dense with foliage all the way up to the canopy line, and even during the dry season humidity levels are much higher than those in Arizona. The fact that higher frequencies will travel further due to this more humid climate may even be completely canceled out because of just how many trees and plants there are. Large sections of this wildlife refuge are also opposite that of the forest. Coffee farming has opened up sections of rolling hills where animals will frequently visit because of the rivers that run through them. Because of this, there is no cure-all answer to these issues, as every region could possibly be different. All of the variables are known, but none of them can be tested without on-site recordings, their resulting frequency profiles, and respective analysis.



### 4. On-site Recording and Analysis

#### 4.1 The recording process

A large portion of this project lies in abundant collections of on-site recordings. Because of the remote location and inability to frequently access highly poached areas, over one-hundred hours of audio were captured over a five day period of fieldwork. These recordings aimed to simulate every possible situation in which a gunshot can occur, as well as document the acoustic ecology of each of these spaces. By doing so, frequency profiles of the landscape can be developed, and accurate 1:1 analysis can be made to report the reliability of the detection process its related code.



Fig 4.1 Zoom H2N recorder placed on a fence post, sealed with a nitrile glove and silica gel packets

First, recordings were required so noise profiles of these landscapes could be developed for every time of the day. For this process, five Zoom H2N<sup>8</sup> recorders were placed each day and captured approximately eight hours of audio. These recorders captured sound at 96kHz to make sure every detail was analyzed. Their locations were marked by GPS, and each contained a description of its surrounding foliage, a timestamp, and its respective weather, including temperature and humidity. Each recorder was placed approximately 200m away from one another, and locations were based upon previous knowledge of where poaching occurred. Because the humidity of Las Alturas throughout the dry season can rapidly increase come nightfall, all recorders were wrapped in thin nitrile surgical gloves and sealed using tape with at least two packets of silica gel inside to keep them dry and operating correctly. Previous tests were performed in Arizona to ensure that the thinnest gloves did not critically alter the incoming sound, or block

out the desired higher frequencies. All recorders were placed on moldable tripods and positioned a few feet off the ground wrapped around thick tree branches or fencing whenever possible. This placement off the ground meant that low rumbling frequencies from passing trucks or the rushing river were less likely to get picked up through the vibration of the tripod legs. An example recorder placement can be seen in Fig. 4.1.

<sup>&</sup>lt;sup>8</sup> Zoom H2n Handy Recorder. (2019, March 29). Retrieved on March, 22, 2019, from

https://www.zoom-na.com/products/field-video-recording/field-recording/zoom-h2n-handy-recorder



#### 4.2 Regional discoveries

After the five days of recording, it was clear through spectral analysis and loudness measurement that the most variance in the sound profiles of these locations came primarily from insects at dusk. In order to develop a general frequency profile of the recordings, iZotope RX<sup>9</sup> was used to look at the FFT in the time domain for the hours of audio. Figure 4.2 displays the overall loudest audio and most variation in frequency content across all the recordings. This hour-long section takes place from about 6 to 7 PM. Throughout this transition into dusk, various species of crickets begin to chirp. These high-frequency chirps occupy most of the sonic space above the 2,700Hz range and can be quite loud when close to the microphone. This is highlighted in Figure 4.2 by the brightness of the orange lines extending along the x-axis. The brighter the color, the more energy there is in that sonic event.



Fig 4.2 (X-Axis: Time) (Y-Axis: Hz (0-48,000)) Spectrogram of an hour long ambient recording in iZotope RX

Towards the right side of the above graph, there is a noticeable increase in the amount of sonic events in the middle of the frequency spectrum (Y-Axis). These newly introduced lines of color represent various cricket chirps at different frequencies. In theory, the more chirps that are introduced, the louder the overall audio file will become. To test this the same recording has been analyzed for loudness and spectral centroid in Sonic Visualizer below.



Fig 4.3 Loudness measurement in Sonic Visualizer of an ambient dusk recording

<sup>&</sup>lt;sup>9</sup> IZotope Inc. (n.d.). IZotope RX 7. Retrieved on January 10, 2019, from https://www.izotope.com/en/products/repair-and-edit/rx.html





Fig 4.4 Spectral Centroid measurement in Sonic Visualizer of an ambient dusk recording

Although the cricket chirps reside at frequencies well above the range observable for the gunshot, there was concern that the louder chirps very close to the microphone would overpower a distant shot, especially during dusk hours. As shown in Figure 4.3, there is a slight increase in loudness over time. These chirps could also negatively affect the spectral centroid. Because the spectral centroid in Figure 4.4 takes in to account the *average* location of energy across the frequency spectrum, if the gunshot is of equal or lesser energy than the chirp in the same frame, the centroid value will not drop as drastically. It is clear near the right side of the spectral centroid graph that the chirps are causing a rise in Hertz values. Another possible concern found when building this frequency profile was the river running through the middle of the land. In some major sections of this river, the water runs rapidly and it is evident in the spectrograms such as Fig. 4.2 that this low rumbling noise could be emitted for hundreds of meters. Just as the energy from the crickets could overpower the gunshots, the rumbling of the subsonic muzzle blast of the gun. It would not be possible to verify whether or not this would hinder detection until gunshots were recorded in these locations.

#### 4.3 The inverse effect of energy

Two of the five days spent collecting audio also involved controlled gunshot collection. During this time two contrasting locations were chosen to simulate likely experiences in which gunshots would occur. These controlled tests included placement of microphones at measured distances facing specific directions, as well as weather documentation, timestamping, and efforts to suspend the units off the ground to emulate their future placement just below the canopy.



Forest recordings were collected first:

Microphone 1 (M1D2) (15m from shot)
Microphone 2 (M2D2) (407m from shot)
Microphone 3 (M3D2) (770m from shot)
Microphone 4 (M4D2) (750m from shot)

The tests were performed in a very dense area of foliage along a path where poaching occurs frequently, due to a



Fig 4.5 Map of recorder locations for forest gunshot testing

public road intercepting private land, as seen at mark M2D2 in Fig. 4.5. It was predicted that the supersonic bullet crack would roll off at a shorter distance than that of the subsonic boom of the muzzle blast. This is evident in the analysis shown in Fig. 4.6. The graph highlights a



Fig 4.6 (770m from shot) Green: Spectral Centroid Purple: Loudness



one-minute section cut from M3D2 at 770m from the point-of-shot. Due to the higher frequency energy of the forests natural sounds, there is a very noticeable and guick drop in spectral centroid (shown in green) from ~5500Hz to ~1700Hz when the gunshot is introduced, and a gradual increase back to its resting centroid following the reverberant crack of the bullet. This is mirrored by an opposite spike in loudness which can be observed in purple. As the microphones are placed closer to the shot the results are even more apparent, this can be observed in Fig. 4.7 which was recorded 15m away. The speed at which these values change remains constant, but the closer to the shot, the larger inverse effect of energy versus spectral centroid is observed. Unfortunately, as the recorders were placed along the security road for ease of access, a 4x4 vehicle passed by Microphones 3 and 4 as the shots were taking place, compromising the audio collection for both those units



*Plains* tests were performed the following day:

Microphone 1 (M1D5) (20m from shot)Microphone 2 (M4D5) (250m from shot)Microphone 3 (M2D5) (610m from shot)Microphone 4 (M4D5) (960m from shot)

Not all poaching occurs in dense forest so a second round of shots was completed in a more open area of the preserve. The recording was also completed at dusk so the ambient loudness of the surrounding



Fig 4.8 Map of recorder locations for plains gunshot testing

area is much higher than the last data gathering session, and a larger number of crickets are audible. Observable changes in spectral centroid and loudness can be seen in all graphs from all four microphones placed. Because of this, it is most important to observe Microphone 3 as it is



nearly 1km away from the shooter, the furthest distance recorded. Not only this, but all tests were performed using a .22 caliber long rifle, the smallest caliber used by poachers. This smaller caliber is the most quiet and least powerful, so if detectable at this distance then any larger caliber will also be detected. Upon listening to the recording the shot is hardly detectable to human ears, but analysis proves numerical evidence that there is a unique drop in spectral centroid with a very steep vector of change.

The difference in spectral centroid is so drastic that if zoomed out to a sixty second clip of the full hour long recording in Fig. 4.8, there are four extremely visible instances where the spectral centroid value drops that is unrivaled by any other sounds.

Fig 4.9 (960m from shot) Green: Spectral Centroid Purple: Loudness



It is important to note that through the one-hundred plus hours of recordings, it was this hour that contained the loudest collection of natural sounds. Even during these loudest points, the spectral centroid responded with a unique and recognizable footprint of every gunshot all the way up to 1km, without any background cancellation or filtering.

#### 4.4 Validation of the vector of change

These controlled gunshot recordings and their respective analysis gave verification that monitoring the vector of change for both spectral centroid and loudness is a viable option for more reliable detection. When combined with the inverse properties of these two metrics, they provide an extra layer of confirmation for a possible shot. Not only has this been verified, but its inclusion has proved that it is also a viable option instead of performing adaptive background subtraction and cancellation. This would free up data and power to fit along the lines originally set forth for this project. The spectral centroid calculation takes into account every bin of frequency and averages it to output the weighted value in Hz. This means that altering the incoming audio before it can be processed would negatively affect the spectral centroid. There is a reliance on the high-frequency crickets to make the spectral centroid variance more drastic, and if filtering was introduced to subtract the low rumble of the river, it would cancel out the necessary frequencies to monitor subsonic shots. This vector of change gives the ability to ignore constant or unchanging sounds, and because the only observable values of difference are from frame to frame, the rumble of the river will not come in to play as it never stops or changes.

While many positive results stemmed from these controlled audio collections, it was also noted that placement of these microphones will play a large role in the natural sounds which they pick up. Because they included plastic tripods wrapped around trees, they are still much closer to the ground then at the proposed canopy-line for the final units. This could have introduced unwanted low-energy into the audio which would be mitigated upon their proper placement.



Fig 4.10 Spectrogram of plains gunshot @960m viewed in iZotope RX



### 5. Building Code

#### 5.1 Why Teensy 3.6

Teensy<sup>10</sup> is a microcontroller development board created by PJRC and designed by the co-founder Paul Stoffregen [24]. Multiple versions of this board exist, each with different speed and memory capabilities, however, all boards utilize the Arduino<sup>11</sup> IDE interface and C/C++ coding language. Two types of Teensy were analyzed and tested for this project, the 3.2, and the 3.6. Research conducted in this project's literature review [17] found that in comparison to other microcontrollers such as the Arduino platform and Raspberry Pi<sup>12</sup> boards, the Teensy is



Fig 5.1: Teensy 3.2 [24]

capable of much higher sampling rate due to a more powerful ADC (Analog-to-Digital Converter). Along with this higher sampling rate, the Teensy platform boasted an average power draw of 45 mA/h for the 3.2, and 90mA/h for the 3.6, significantly less than that of the two other platforms.

The Teensy 3.2 was chosen for initial development due to an optional Audio Board add-on available through the PJRC website. This board allows the computer to access the Teensy device as audio output. By doing so, audio can be passed through the board to be analyzed in real-time, instead of preloading & running the files from a micro-SD card. This was necessary as the amount of audio collected on-site for analysis was very large, making the transfer to an SD card not possible for more than one file at a time. This playback through the device also simulates the exact

conditions under which a microphone

would be connected to the unit and listening. A very helpful component which the Teensy board features is their Audio Library and Audio System Design Tool. The Audio Library features an extensive set of functions for recording, analysis, mixing, and more [25]. To help users learn this library, the Audio System Design Tool was created. This design tool is a visual programming space which allows users to drag, drop, and connect features from the Audio Library with one another to build the framework for the desired project. Once all features desired are added, an export function creates and



Fig 5.2: Teensy 3.2 with attached Audio Board

<sup>&</sup>lt;sup>10</sup> PJRC. (n.d.). Teensy. Retrieved on April 10, 2019, from https://www.pjrc.com/teensy/

<sup>&</sup>lt;sup>11</sup> Arduino. (n.d.). Retrieved on January 14, 2019, from https://www.arduino.cc/

<sup>&</sup>lt;sup>12</sup> Foundation, R. P. (n.d.). Raspberry Pi. Retrieved on April 10, 2019, from from https://www.raspberrypi.org/



copies code which can be directly pasted into the Arduino IDE. This code contains all the necessary setup and pin distribution for the Teensy so that the audio board can be used right away, along with the functions included. Due to this ease of code design through the audio board and Audio System Design Tool, the 3.2 was a good starting point to test the on-site audio. However, it is noted that a major component lacking in the 3.2 but included in the 3.6 is a real-time clock. In order to compute time-difference on arrival of the shots once detection was verified, an accurate clock must be included on the board. For this reason, the 3.6 platform was ultimately chosen to replace the 3.2 for a future implementation including localization.

#### 5.2 Initial MATLAB algorithm principles

The use of the LibXtract toolkit within Sonic Visualizer provided sufficient visualization of spectral feature extraction, allowing for positive identification of the inverse energy and spectral centroid theory proposed in chapter two. However, before beginning to build this code in C/C++ and the Arduino IDE, it was necessary to compare the Sonic Visualizer output to output from an industry standard program to verify correctness.

For this reason, MATLAB<sup>13</sup> was chosen to perform FFT and feature extractions, and the associated graphs were compared to those generated within Sonic Visualizer. Simulink's "Audio Toolbox" is a widely trusted set of tools for performing these extractions. The first of these extractions regarded the performance of an FFT. All related code regarding MATLAB FFT can be found in Appendix B. This code receives various inputs as laid out in chapter two to create an FFT graph from an audio file, the graphs created can be viewed in Figures 2.2 and 2.3. Initial MATLAB code to perform loudness and spectral centroid calculation was provided with the help of Walter Zimmer [25]. This code can be viewed in Appendix C, and allowed for accurate plotting of feature extraction points. It was through these tests within MATLAB that the distinction and decision to choose energy over loudness was made. The mathematical calculation to convert the energy of a signal to the psycho-acoustic parameter loudness involves another level of multiplication in order to better represent what human ears perceive. This calculation is not useful for purposes of this project as the energy metric provides sufficient information.

#### 5.3 Calculating the FFT and energy

A key analysis component of the Teensy Audio System Design Tool features a 1024 point FFT component. Applying this component in the design tool interface builds code that prepares the Teensy board to perform this FFT on audio data played back by a medium of choice, this can include the available micro-SD card slot, or directly as the computer output. The output of this module includes 512 frequency bins



Fig 5.3 The FFT object within Teensy's Audio System Design Tool

<sup>&</sup>lt;sup>13</sup> MATLAB. (n.d.). Retrieved on January 13, 2019, from https://www.mathworks.com/products/matlab.html



each with approximately 43hz of data per bin. Each of these bins reports its respective energy eighty-six times a second, and multiple bins can be grouped together or averaged [26]. This can be useful to keep processing power low, by averaging the groups of frequencies deemed unnecessary for the application. By writing these energy values to an array every frame of calculation, a spectrum of all 512 bins can be created. For purposes of low power consumption, an array of twenty values was created for this project, and the less important frequencies above 1500hz were combined together and averaged in groups of 10's, 50's and 100's. This division of bins allows for a higher frequency resolution in the sub-1500hz region, frequencies that will be relied on for energy analysis of the subsonic gunshot. These divisions of bins can be viewed in the primary bulk of code for this project located in Appendix D. Before being able to calculate the vector of change, the difference in energy must be noted. It was discovered during this process that although all 512 bins of the FFT analysis must be computed in order to complete the spectral centroid following the energy analysis, it is not necessary to use its respective twenty energy values written in the array. For example, it is possible to only pull the first six values for energy, essentially allowing for the energy to be measured in the Ohz to 1500hz range. This process bypasses the need of any low-pass filtering. In order to calculate the difference from frame to frame, values of the array are summed and averaged, then subtracted from the previous frames total. The code below displays the first 10 bins being siphoned into a six value array named "level."

```
level[0] = myFFT.read(0);
level[1] = myFFT.read(1);
level[2] = myFFT.read(2);
level[3] = myFFT.read(3, 4);
level[4] = myFFT.read(5, 6);
level[5] = myFFT.read(7, 8);
level[6] = myFFT.read(9, 10);
```

Upon completion of this process, the current energy is written in to the variable "previous energy," and as the process begins again this keeps an up to date difference in energy, eighty-six times per second. This energy difference value is then stored within a variable to be used during the vector of change calculation.

#### 5.4 Calculating the spectral centroid

Mathematical computation of the spectral centroid revolves around the FFT calculation and application of the equation shown in chapter 2, section 4. Appropriate representation of the centroid relies on an unfiltered audio input, resulting in all twenty values written to the array from the FFT calculation being used. As previously stated, higher frequency energy will need to be present in order to see a drop in centroid upon the arrival of the subsonic waves to



the microphone. To calculate this value, the energy reported in each bin, or group of bins, is multiplied by its mean Hertz value. This means that for bin 0 which is represented as 0hz to 43hz, the energy value would be multiplied by 21.5hz. This process occurs for every value in the array separately. Once calculated, all respective array values are summed, and then divided by the summed value of energy for that frame. This calculation outputs a value in Hertz which represents the weighted average of energy in that frame. Once more, the calculation of this parameter written in C/C++ can be viewed in Appendix C. While the spectral centroid value in Hertz is kept as a necessary variable which will be analyzed with a threshold, the difference calculation must also be computed similar to energy, so that the vector of change for the spectral centroid can also be calculated. This is performed in the same manner, by subtracting the current centroid value from the previous frame's.



#### 5.5 Vector math in C/C++

Once difference values for both the energy and spectral centroid are calculated it is possible to analyze the vector of change for both variables. Using the equation displayed in section 2.5, the magnitude value for energy can be calculated in the code as such:

```
hyp = (sqrt((pow((adj), 2) + (pow(diffLevelAvg, 2)))));
```

The variable "hyp" in this instance is the hypotenuse (c) of a right triangle, while "diffLevelAvg" is the opposite side (b) and "adj" refers to the adjacent side (a). This can be further explained by the Pythagorean Theorem:



Because this code is being called 86 times per second, the value "adj" will always be a constant. For purposes of continuity, the variable is declared as 1024. Because the opposite (diffLevelAvg) is calculating from frame to frame, this value represents the energy level difference of the current frame minus the previous. This final equation can be written as:

$$\begin{vmatrix} \overrightarrow{AB} \end{vmatrix} = \sqrt{\left(x_2 - x_1\right)^2 + \left(y_2 - y_1\right)^2}$$
$$hyp = \sqrt{adj^2 + diffLevelAvg^2}$$

This equation will return the magnitude of the desired value. The same equation can apply for both energy and spectral centroid, as long as the respective difference value is input for opposite (b) as shown below.



Once the magnitude is calculated, the direction vector may be derived. This value will return the angle difference from frame to frame of both energy and spectral centroid. This is displayed in the example below as theta.



Mathematically this calculation for energy can be written as:

$$\theta = \tan^{-1} \left( \frac{diff LevelAvg}{adj} \right)$$

This is presented in the code as:

```
thetaRad = (atan(diffLevelAvg / adj));
```

This equation will output the angle vector in radians, so a further step is required to convert this value to degrees:

```
thetaDeg = (thetaRad * (180/pi));
```

The resulting output will provide the angle vector in degrees for energy, and can be calculated for spectral centroid by replacing the "diffLevelAvg" with the difference in spectral centroid, "diffCentroid." *The resulting degree value of spectral centroid will only be important when it's reported as negative, as it's the sudden drop in centroid that is being monitored.* 

```
SCthetaRad = (atan(diffCentroid / adj));
SCthetaDeg = (SCthetaRad * (180/pi));
```



#### 5.6 Parameter verification & shot detection

Upon completion of spectral feature extraction, there are five components which will be monitored. When all conditions are met and the threshold is passed, a report of "Shot detected" will be sent to the serial port along with these variables. These variables include:

#### Magnitude:

```
Energy Difference = hyp
Centroid Difference = SChyp
```

#### Direction:

Energy Angle = thetaDeg
Centroid Angle = SCthetaDeg

#### Spectral Centroid (in Hz):

Spectral Centroid = SpectralCentroid

An "if" statement is used to verify the shot and tune the thresholding of each parameter. These thresholds must be tested and tuned to fit the requirements needed to accurately detect a shot, while ignoring unwanted sounds.

```
if ( (hyp > 1020.0) && (thetaDeg > 7.0) && (SChyp > 2100.0) && (SCthetaDeg < -50) && (SpectralCentroid < 5000) )
```

The inclusion of spectral centroid on it's own adds a necessary layer of detection. Because the vector calculations are solely comparing frame to frame, there must be a variable which monitors whether or not these changes are occurring at the lower-frequencies of a gunshot, and not the ranges of crickets or other insects which may result in a false positive.

An example report of a shot to the serial monitor is seen as:

```
SHOT DETECTED
Energy Difference: 1049.17, Centroid Difference: 3091.34
Energy Angle: 12.58, Centroid Angle: -70.66
Spectral Centroid: 4060.77hz
```



## 6. Final Testing & Results

#### 6.1 Accuracy of detection

In order to measure accuracy of detection, a host of tests from gunshot recordings at several distances were played through the Teensy 3.2 via means of the audio output from the computer. Each of the compositions included 100 shots from every distance to replicate one-hundred shots that may occur in the field. In order to test reliability, only *one* set of thresholds was created that would be used for all distances. Strenuous tuning of the system before these tests proved that there is no simple answer to fulfill all needs. Two locations were tested, the plains, and the forest of Las Alturas del Bosque Verde in Costa Rica.

Distance	20m	250m	610m	960m	TOTALS
Total Detections	104	102	100	97	97.75%
False Positives	4	2	0	0	6
Missed Detection	0	0	0	3	3

#### Plains test location (out of 100 total shots)

It was evident through testing that a more sensitive set of thresholds favored quieter shots, recorder further from the source, but was more prone to false positives during closer shots (250m meters or less), as amplitude levels extended through multiple frames due to reverberation at close distance. Although these recordings attempted to take into account all variables, they were not perfect. For one, all recorders mounted to tripods were still subject to low-frequency vibrations being carried through the tripod's legs, causing extraneous energy and unwanted spikes in amplitude during closer shots. It's only through placement higher up in the forest canopy that this issue would completely be resolved. For this reason, a more sensitive set of thresholds was chosen to provide accurate detection at long ranges, while risking a few false positives as a trade-off. It should also be noted that once these units are placed in the canopy, the likelihood of a shot occurring at 20m is very low due to the large areas of monitoring desired, and it would be wiser to prepare the units for softer shot detections. Lastly, all false positives occurred in the frame following a gunshot due to amplitude values lasting more than



one frame, and none were caused by the natural sonic environment.

This issue of microphone placement seemed even more troublesome for the forest recordings. As stated in chapter 4, section 3, it is noted that a 4x4 vehicle driving on the adjacent road compromised the shots of the most distant Microphone 4. During the final tests, it became apparent this same 4x4 vehicle also compromised the audio collected in Microphone 3. The energy of the vehicle carried to Microphone 3, 120m east of Microphone 4, and masked the very soft footprint of the gunshot occurring 770m away. This was also a product of the naturally dense forest environment not allowing sounds to travel as far due to the absorption of dense foliage.

Distance	15m	407m	770m	750m	TOTALS
Total Detections	109	103	-	-	94%
False Positives	9	3	-	-	12
Missed Detection	0	0	-	-	0

#### Forest test location (out of 100 shots)

Error Rate	9%	3%	-	-	6%

Results from these controlled tests show that the current detection algorithm with a single set of thresholds reports an accuracy of 97.75% up to 960 meters in the plains, and 94% up to 407 meters in the forest. The reports also display the need for a specific distance from the service road upon final placement in order to mitigate road noise masking the gunshot sound. Although vehicles accessing this road is very uncommon, it can block the incoming energy from gunshots up to 120m from the vehicle. Further testing with vehicles and the road would need to occur before concluding with the optimum distance from the road to minimize undesired sound masking.



### 7. Future Considerations

As the localization portion of this project is an ongoing endeavor, considerations must be made to further implement the code and results presented in this paper to properly fit within the realm of wireless communication, and provide evidence against those hunting illegally. To this the, the following outlines future work, ongoing in the Acoustic Ecology Lab at ASU.

**LoRa**<sup>14</sup> **line-of-sight:** Extensive testing will need to be completed to verify the wireless distance communication of the "Long Range" modules. These distance capabilities coupled with the detection tests in chapter six will ultimately decide how far apart the microphone units will be from one another to transmit data and accurately recognize gunshots on multiple units. Initial work shows good reception, and introductory integration has been completed with the gunshot detection system.

*Low data:* With five detection variables currently being output by the algorithm, decisions must be made to choose the more important (or none) variable to be shared by the wireless transmitter on each microphone unit. Long distance communication of LoRa modules over 20km relies on small packet sizes of just a few bytes to accurately send and receive data.

**Real-time clock:** The algorithm must be reorganized and revised to disregard the Teensy 3.2 and its audio shield, and work with the Teensy 3.6 platform with an attached i2S MEMS digital microphone. The real-time clock feature on the 3.6 is vital for localization, and timestamps must be sent out gunshot detection along with a unit identification number for each system with a positive gunshot detection.

*Microphone protection:* Similar to the protection of the Zoom H2N when performing test recordings on-location in Costa Rica, final microphone and wireless transmitter units must be built in sealed cases to protect electronic components from moisture. These cases must also not block out necessary acoustic energy from reaching the microphone itself while managing heat-related challenges.

**Buffer recording:** If reliable detection and localization of a specific gunshot lead to the capture of criminal poaching, audio taken from the recording device and stored on the micro-SD card could prove to be sufficient evidence in a court of law. For this reason, a future

<sup>&</sup>lt;sup>14</sup> What is LoRa®? (n.d.). Retrieved from https://www.semtech.com/lora/what-is-lora



consideration will likely involve a buffer of audio being saved until detection is complete. When the detection comes back positive, this small recording will be time-stamped and saved to the SD card on the unit for further analysis if the poachers are captured. If the detection comes back negative, the audio buffer will be deleted and it will continue to record frame to frame.



### 8. Conclusion & Acknowledgements

The procedures detailed in this paper have laid out the foundation of a low data algorithm which utilizes spectral feature extraction and monitoring as a means of gunshot detection. Future considerations will continue to utilize and build upon the algorithm created to accurately localize, alert, and provide assistance to the security detail of this protected region. The on-site recordings and accuracy results of this algorithm have provided a confirmation of reliability for gunshots up to the tested range of 960 meters from the source in the plains region, and 407 meters in the forested region of Las Alturas de Coton in Costa Rica. Findings have yielded insight into the acoustic responses of this specific environment and their influence on the outlined analysis approach. Through further testing of other locations, it is believed the spectral parameters and code used for this specific region could be successfully generalized and transferred to other site-specific applications.

I'd like to take this section to thank all those who have helped to make this project what it has become. This process has taught me the importance of friends, peers, and most importantly that the desire to complete an objective which you truly believe is for the greater good will make the countless hours spent to reach it always worth-while. If it were not for those mentioned and their willingness to aid and support, this project would not be where it is today.

A special thank you to Dr. Garth Paine for his mentorship and continued support throughout the entire research process and beyond. Your prior work and teachings in the field of sonic studies and specifically acoustic ecology has taught me to approach the sonic environments we live in with a keen ear. While numbers may give to us verification of our theories or data of which to present, your wisdom has shown me that there is nothing more powerful than just stopping and listening.

To Dr. Sabine Feisst for her encouragement and aid for the entire process and more, your mentorship within the Acoustic Ecology Lab provided the groundwork to pursue this endeavor.

To Dr. Robert LiKamWa for his willingness to provide technical aid at a moments notice.

To Walter for your countless hours of help in the creation and testing of code, and different means to bring the spectral parameters discussed to life on the Teensy platform.

Thank you to the entire conservation team at the Phoenix Zoo and Las Alturas del Bosque Verde. Jan Schipper for his faith, enthusiasm, and openness to provide the resources and opportunity to travel abroad and conduct this research with his team. Chelsey, Andy, Roci, Fernando, and the entire team for their help to capture recordings, and for always being there for support in the process. To Karl and Judy for their unwavered hospitality while abroad.

Lastly, I'd like to thank my loved ones, Tracy, Jeff, Cameron, and Nikki for the outpouring of constant support and patience through this process.



### References

[1] R. C. Maher, "Acoustical characterization of gunshots," in Signal Processing Applications for Public Security and Forensics, 2007. SAFE '07. IEEE Workshop on, pp. 1-5, april 2007.

[2] R. C. Maher, "Modeling and signal processing of acoustic gunshot recordings," in Proc. IEEE Signal Processing Society 12th DSP Workshop, 2006.

[3] A. K. Bandi, M. Rizkalla and P. Salama, "A novel approach for the detection of gunshot events using sound source localization techniques," *2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS)*, Boise, ID, 2012, pp. 494-497.

[4] A. Chacon-Rodriguez, P. Julian, L. Castro, P. Alvarado and N. Hernandez, "Evaluation of Gunshot Detection Algorithms," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 2, pp. 363-373, Feb. 2011.

[5] B. Sadler, L. Sadler, and T. Pham, "Optimal and robust shockwave detection and estimation," in *Proc. IEEE ICASSP*, 1997, vol. 3, pp. 1889–1892.

[6] C. Clavel, T. Ehrette and G. Richard, "Events Detection for an Audio-Based Surveillance System," 2005 IEEE International Conference on Multimedia and Expo, Amsterdam, 2005, pp. 1306-1309.

[7] K. (1970, January 01). **Rural hacker**. Retrieved February 01, 2018, from http://ruralhacker.blogspot.pt/2013/01/arduino-micloc-or-acoustic-location.html

[8] Rural hacker. (n.d.). Retrieved February 01, 2018, from http://ruralhacker.blogspot.com/p/micloc.html

[9]https://www.amazon.com/SparkFun-LYSB0110I2IMW-ELECTRNCS-Teensy-XBee-Adapter/dp/B0110I2IMW

[10] Shovic, J. C. (n.d.). **Solar Power Management** » Raspberry Pi Geek. Retrieved February 10, 2018, from <u>http://www.raspberry-pi-geek.com/Archive/2015/10/Managing-solar-power-systems-with-SunAir-boards</u>

[11] **Thread: Achieving Low Power Consumption with Teens 3.1 and the latest Teensyduino Version.** (n.d.). Retrieved February 10, 2018, from

https://forum.pjrc.com/threads/28122-Achieving-Low-Power-Consumption-with-Teens-3-1-and-the-latest-Teensyduino-Versio

[12] XBee Buying Guide. (n.d.). Retrieved February 10, 2018, from https://www.sparkfun.com/pages/xbee\_guide

[13] Sue, R. &. (n.d.). Vacation Planning. Retrieved February 10, 2018, from <u>https://costa-rica-guide.com/travel/weather/sunshine-temperature-map/</u>

[14] **Teensy 3.1 XBee Adapter.** (n.d.). Retrieved February 10, 2018, from <u>https://www.amazon.com/SparkFun-LYSB0110I2IMW-ELECTRNCS-Teensy-XBee-Adapter/dp/B0110I2IMW</u>

[15] XBee. (2017, January 31). Retrieved February 10, 2018, from https://en.wikipedia.org/wiki/XBee



[16] MATLAB. **GCC-PHAT Documentation**. Retrieved January 7, 2019, from https://www.mathworks.com/help/phased/ref/gccphat.html?s\_tid=gn\_loc\_drop

[17] Arar, S. (2017, August 28). An Introduction to the Fast Fourier Transform. Retrieved March 15, 2019, from https://www.allaboutcircuits.com/technical-articles/an-introduction-to-the-fast-fourier-transform/

[18] **Sampling (signal processing).** (2019, February 25). Retrieved March 15, 2019, from https://en.wikipedia.org/wiki/Sampling\_(signal\_processing)

[19] Windows and Spectral Leakage. (2019, January 03). Retrieved March 17, 2019, from https://community.plm.automation.siemens.com/t5/Testing-Knowledge-Base/Windows-and-Spectral-Leakage/ta-p/432760

[20] Bullock, J. (n.d.). LibXtract. Retrieved March 17, 2019, from https://code.soundsoftware.ac.uk/projects/libxtract

[21] Sengpiel, E. (n.d.). **The Human Perception of Loudness.** Retrieved March 19, 2019, from <u>http://www.sengpielaudio.com/calculator-loudness.htm</u>

[22] Spectral centroid. (2018, May 27). Retrieved March 13, 2019, from https://en.wikipedia.org/wiki/Spectral\_centroid

[23] Sweetwater. Effects of Temperature and Humidity on Live Sound. Retrieved March 17, 2019, from <a href="https://www.sweetwater.com/insync/effects-of-temperature-humidity-live-sound/">https://www.sweetwater.com/insync/effects-of-temperature-humidity-live-sound/</a>

[24] Sam. (2017, November 13). **What is Teensy?** - Tutorial. Retrieved March 20, 2019, from <u>https://core-electronics.com.au/tutorials/what-is-teensy.html</u>

[25] Zimmer, W. (n.d.). Spectral feature extraction example code in MATLAB. Retrieved February 13, 2019.

[26] **Teensy - Audio System Design Tool.** (n.d.). Retrieved April 1, 2019, from https://www.pjrc.com/teensy/gui/?info=AudioAnalyzeFFT1024

[27] Chioye, L. (2013, December 06). **Choose The Right FFT Window Function When Evaluating Precision ADCs.** Retrieved April 5, 2019, from <u>https://www.electronicdesign.com/analog/choose-right-fft-window-function-when-evaluating-precision-adcs</u>

[28] Hoefer, K. (2018, January 20). Real-time Gunshot Detection and Localization for Environmental Protection. Retrieved, from www.kylehoefer.com

[29] Bullock, J. (2008). *Implementing audio feature extraction in live electronic music*. Birmingham City University. Retrieved March 20, 2019, from <a href="https://www.academia.edu/4493811/Implementing\_audio\_feature\_extraction\_in\_live\_electronic\_music">https://www.academia.edu/4493811/Implementing\_audio\_feature\_extraction\_in\_live\_electronic\_music</a>

[30] Paine, G. (2017). "Acoustic Ecology 2.0". Contemporary Music Review (15, Nov. 2017)

[31] Paine, G. Listening to nature: How sound can help us understand environmental change: The Conversation, December 21, 2018 <a href="https://theconversation.com/listening-to-nature-how-sound-can-help-us-understand-environmental-change-105794">https://theconversation.com/listening-to-nature-how-sound-can-help-us-understand-environmental-change-105794</a>



### **List of Figures**

**2.1:** Frequency spectrum of sine wave aligning with frequency resolution (red) and sine wave not aligning with frequency resolution (green) [19]

**2.2:** FFT of gunshot and length 1024

2.3: FFT of gunshot at length 16,384

2.4: Loudness analysis in Sonic Visualizer, over a period of three gunshots

2.5: Spectral centroid analysis in Sonic Visualizer, over a period of three gunshots

**2.6:** Spectral centroid (RED) and Loudness (PURPLE) analysis in Sonic Visualizer, over a period of three gunshots

3.1: The initial testing location in Four Peaks, Arizona

3.2: Shots were recorded with a .223 bolt-action rifle

4.1 Zoom H2N recorder placed on a fence post, sealed with a nitrile glove and silica gel packets

4.2 (X-Axis: Time) (Y-Axis: Hz (0-48,000)) Spectrogram of an hour long ambient recording in iZotope RX

4.3: Loudness measurement in Sonic Visualizer of an ambient dusk recording

4.4: Spectral Centroid measurement in Sonic Visualizer of an ambient dusk recording

- 4.5: Map of recorder locations for forest gunshot testing
- 4.6: (770m from shot) Green: Spectral Centroid Purple: Loudness
- 4.7: (15m from shot) Green: Spectral Centroid Purple: Loudness
- 4.8: Map of recorder locations for plains gunshot testing
- 4.9: (960m from shot) Green: Spectral Centroid Purple: Loudness
- 4.10: Spectrogram of plains gunshot @960m viewed in iZotope RX
- 5.1: Teensy 3.2
- 5.2: Teensy 3.2 with attached Audio Board
- 5.3: The FFT object within Teensy's Audio System Design Tool



## Appendix

#### Appendix A - Literature Review

Real-time Gunshot Detection and Localization for Environmental Protection Literature Review

> Kyle Hoefer January 20th, 2018 Arizona State University

#### Abstract

The ability to detect and triangulate specified acoustic sound sources using microphone arrays as a tool for surveillance has become a critical piece of military and civilian protection. These systems are often extremely power consuming, and require a dispatch team to verify the alerts before responding [3]. The goal for this project is to create multiple self-sustaining solar powered gunshot detection units to aid anti-poaching patrols on a Costa Rican wildlife preserve. These microphone array units will utilize real-time spectral feature extraction to distinguish gunshots against the rainforest landscape, calculate TDOA and DOA to each unit, and wirelessly relay shooter coordinates and direction to ground patrol. Key aspects of this project include a completely autonomous system from detection to dispatch and a reliable processor which can operate for up to three months at a time without maintenance.

#### Process

Detection: Reliably distinguishing a gunshot against a natural soundscape will vary greatly depending on the terrain, weather conditions, type of firearm used, and distance from and direction of the shooter. As stated by Robert Maher in [1][2], firearms will typically create three sonic events upon discharge. These can be classified as mechanical action, muzzle blasts, and shockwaves. Mechanical action in this instance refers to the automatic cocking mechanism found on many handguns and semi automatic rifles, which can be heard up to only a few meters away, rendering it useless for this caliber of detection. Muzzle blast describes the energy emitted from the front of the firearm. This event lasts typically 3-5 milliseconds and is stronger when facing the shooter, although the energy wave is dispersed spherically at the speed of sound. Shockwaves are in reference to the bullet reaching or surpassing the speed of sound. These waves typically last 200 microseconds and propagate outwards from the bullets path at its highest speed, becoming increasingly parallel to the bullet as it begins to slow [3]. Although amplitude variation will occur depending on direction of the shot, shockwaves will always reach a specific location prior to the muzzle blast if the bullet surpasses the speed of sound. There are a few calibers of ammunition which operate subsonic, however these rounds are rarely used when hunting large game so they will be of minimal concern for this project.



When distinguishing distant gunshots from closer naturally occurring sounds such as rain, falling objects, animals, etc, it is unreliable to look at amplitude measures, even if these measures are taken at specific frequencies [4]. However, the initial rise time of the shockwave is less than 2 microseconds, much quicker than any of the natural sounds listed above. You may look at this rise time of any sound dependent of the amplitude as the primary detection method, as long as the acoustic impulse response of the surrounding area is taken in to consideration. Given a long enough frame time to analyze the short time energy of the wave, examples from Clavel in [6] report that analysis of the first eight Mel-Frequency Cepstral Coefficients (MFCC) along with the first two spectral statistical moments, namely spectral centroid, can provide a very robust feature extraction set for training a system that looks at "shot" versus "no shot" scenarios. Chacon-Rodriguez et al. and Sadler et al. report in [4, 5] that these N-shaped shock waves can lose their shape due to nonlinear dispersion, falling below supersonic speed, or hitting an obstacle, which is a higher possibility in a dense setting such as a rainforest. Even so these N-shaped shock waves are more useful in distinguishing the differences between discharged firearms and not necessarily a general consensus on whether it was a gunshot or a naturally occurring sound, meaning these risks in this project are mitigated. With this being said a proper first approach to detection would begin with the MFCC's and spectral centroid analysis given by Clavel's example. As Maher also states in [2], because of this shockwave rise time, typical 48khz audio sample rates will not be sufficient enough to detect this extremely quick change even though relative shockwave arrival times can still be deduced using multiple sensor channels. This means a sample rate of 96khz or higher may be required in order to pick up proper detection before moving on to localization.

*Localization:* Bandi *et al.* report techniques for localization in [3] in conjunction with muzzle blast, shockwave, and constant analysis of air temperature. Because the speed of sound in air increases with temperature, each microphone array will need to be mounted with a temperature sensor to report variables before the detection algorithm can be run. Any discrepancy in temperature changes between units can report a false direction and compromise the detection.

$$c = c_o \sqrt{1 + \frac{T}{273}}$$

In this instance, *c* represents the speed of sound in the air, while  $c_o = 331 \text{ m/s}$  represents the speed of sound in the air at 0 degrees Celsius. *T* is the temperature at that instance in Celsius when the detection of a possible gunshot was determined [3]. Although this report is based on distinguishing different types of firearms at these ranges, this equation will still be necessary for microphone array units.

Bandi also reports that a minimum of two microphones on the unit is enough to find the direction of-arrival (DOA) estimation, however you will need three or more to triangulate source localization and gain an accurate idea of how far away the shot is. This statement holds true if



you are only operating one unit, however in the context of this project there will be multiple units all reporting at the same time. It will still be beneficial to create a robust model with more than two microphones, however if computational data must be sacrificed to improve power consumption two microphones will suffice as each unit will display a direction converging on a point.

To gather DOA you must first find the time-difference on arrival (TDOA). Maher and Bandi both state in [2][3] that TDOA estimations such as Maximum Likelihood (ML) and Smoothed Coherence Transform (SCOT) are effective when the noise source is uncorrelated, but when reverberance is included they both failed due to its presence. This being said, the Generalized Cross- Correlation Phase Transform (GCC-PHAT) proved robust in both computational efficiency and presence of noise [3]. "In the example below, *m* and *n* represent the two microphones on the unit, while  $X_m(f)$  and  $X_n(f)$  are the Fourier transforms of the two signals and [] \* denotes the complex conjugate."

$$\widehat{G}_{PHAT}(f) = \frac{X_m(f)[X_n(f)]*}{|X_m(f)[X_n(f)]*|}$$
 [2]

$$\widehat{R}_{PHAT}(d) = \int_{-\infty}^{\infty} \frac{X_m(f)[X_n(f)]*}{|X_m(f)[X_n(f)]*|} e^{j2\pi ft} df \quad [2]$$

The resulting time difference of the two microphones will be estimated as,

$$\widehat{d}_{PHAT}(i, j) = \operatorname{argmax}_{d}(\widehat{R}_{PHAT}(d)) \quad [2]$$

Once completed we jump to the DOA calculation. As stated earlier, a two-microphone system will provide an accurate direction of source detection but this process will become more robust when including additional microphones to the array to present full source localization. Mathematical calculations for DOA via a two-microphone unit are simple and computationally light, utilizing the speed of sound calculation to provide an accurate result each time a possible shot is detected. The figure below is in reference to [3], using the first microphone to receive the waveform as the basis for the second microphone to calculate the time difference and direction.





In this equation, m1 and m2 are the separate microphones on the unit, while  $d(\cos\theta)$  represents the distance that the wavefront must travel before reaching the second microphone. Time difference between the two sensors can be calculated by  $m_{12} = d \cos \theta / c$  where *c* is the speed of sound.

However Bandi speaks in regards to a single array system calculating the localization, and not a mesh network of units. Multiple units reporting a direction variable of a specific area on a 2D

plane can create an area of interest where the sound source may have been created. In this instance it may not be beneficial to include more microphones per unit, rather more units to the mesh network. Illustrated is an instance where 12 units are deployed to monitor a location, the indicated overlap provides a viable "area of interest" where the gunshot could have originated.



Network: Once a signal has been received and processed by the unit confirming a gunshot, the



information regarding the direction and location of the shot must be communicated to a central hub where all mesh network information is sent. The resulting display will direct teams to a



location which they must search. User "Kripthor" on his blog "rural hacker" [7], describes a project from which he created an acoustic gunshot location system using both Arduino and Teensy platforms, along with an nRF24L01 2.4GHz Radio/Wireless Transceiver. This information was sent through geographical position data to a KML file, and networked to Google Earth. The image displayed is from Kripthor's blog post, showing the accuracy of 3 shots detected by the system.

Although precision and distance may not be

entirely accurate, the system only uses one unit to display locations. Because of this the use of trilateration is necessary. However given multiple units communicating at the same time, trilateration may be unnecessary as only a drawn line in the direction of the shot will connect with other lines drawn from other units in the mesh network. Another setback to this instance of localization is that the transceiver used may only operate up to 1km in range, this is much less than the desired range for autonomous monitoring in the Costa Rican preserve. This issue may be solved with use of the XBee wireless platform. XBee transceivers may send and receive data, allowing for a mesh network to be easily created and communicate with one another. By using the "reference" XBee as a home base connected to a computer, this may receive

directional data sent by the units in the field. Models such as the XBee-PRO XSC allow for a outdoor line of sight to be up to 28 miles with external antenna [9], and can communicate with Teensy and Arduino or Raspberry Pi boards using specific adapters [14]. With this combination, sample rates of up to 122 khz without modification and 1230 khz with modification can be fed in to the Teensy board [8], processed through TDOA and DOA analysis, then relayed as coordinate directions via XBee to the main hub once confirmed as a possible shot.





*Power:* A large portion of consideration for this project lies in supplying enough constant power to run these units autonomously for months at a time. There are various combinations of solar power and LiPo battery options available depending on the base board chosen. Basic mA power consumption range from...[10][11]

#### Raspberry Pi (w/ Wireless dongle) (mA)

Model A	Model A+	Model B	Model B+
260	200	480	300

#### Arduino (mA)

Uno	Micro	Mega 2560
50	20	70

#### Teensy (mA)

3.1 Platform	
20	

"Raspberry-pi Geek" [10] runs a good demonstration the amount of solar power and battery power needed to run a Pi for up to 14 hours through the day. This demo assumes 280mA on average with power being delivered at up to 85% efficiency, 6,600mAh LiPo batteries, and two solar cells with a total of 6.8w, and an *average of 8 hours of sunlight per day*. This is a good benchmark demonstration for the Pi alone. However with the required wireless distance transmission of the XBee, an equal if not larger source of power must be supplied to this component. [12]

#### **XBee Power Consumption**

XBee Pro 60mW Wire Antenna - Series 1	XBee Pro 900 RPSMA	XBee Pro 900 XSC RPSMA	XBee Pro 900 XSC S3B Wire
215mA@ 3.3v	210mA@ 3.3v	256mA@ 3.3v	215mA@ 3.3v
1 Mile	6 Miles	15 Miles	28 Miles
250kbps	156kbps	1.6kbps	10-20kbps

Combine the values above with possible Pi combinations and you have a power hungry system in regards to solar power. This is obviously dropped when considering the Teensy platform, however running the necessary algorithms in real time while simultaneously recording a buffer



on that platform at an extremely high sample rate may increase the power consumption extensively.

## All of these numbers pale in comparison to the fact that Las Alturas is located in a region of Costa Rica that receives an average 50% chance of sunlight per *year.* [13]

Necessary first steps in all four sections; detection, localization, network, and power, require a viable platform for coding and prototyping. Communication with a main hub through wireless transmission will also be required before the code behind detection and localization can be addressed. It's through these tests that power consumption will need to be documented and accounted for when equipping units with both batteries and solar power, and appropriate microphone unit housing designs can be considered.



Appendix B - Comparison of FFT window types [26]





#### Appendix C - Spectral feature extraction code in MATLAB [25]

```
%froot='\MacintoshHD\Users\PlacePathHere';
%fname=[froot 'AUDIOFILE.WAV'];
info=audioinfo('AUDIOFILE.WAV');
fs=info.SampleRate;
nch=info.NumChannels;
for ii=4.5
  xx=audioread('AUDIOFILE',ii*60*fs+[1 60*fs]);
  figure(1),plot(xx)
 title(sprintf('%d',ii))
  drawnow
end
st=128; %step size
fd=fs/st; %resulting sampling frequency
nn=(size(xx,1)-1024)/st; %number of samples in output vector
Y1=zeros(nn,1); %result vector 1
Y2=Y1;%result vector 2
N=(3:256)'; %frequency bins to use (always ignore bin 1 or DC)
for jj=1:nn
    % if mod(jj,1000)==0, jj, end
    j1=1+(jj-1)*st;
    j2=j1+1023;
    uu=xx(j1:j2,1).*hann(1024);
    yy=fft(uu,1024);
    vv=(abs(yy(N)));
    Y1(jj)=sum(vv.*(N-1))/sum(vv); %centroid
    Y2(jj)=sum(vv.^2); %intensity
end
n2=100;
Y3 = filter(ones(2*n2,1)/(2*n2),1,[Y2(1)*ones(n2,1);Y2]); Y3(1:n2)=[];
Y4 = filter(ones(2*n2,1)/(2*n2),1,[Y1(1)*ones(n2,1);Y1]); Y4(1:n2)=[];
SNR1=10*(Y4./Y1-1);
SNR2=(Y2./Y3-1);
figure(2),plot((1:nn)/fd,SNR1,(1:nn)/fd,SNR2)
figure(3),plot((1:nn)/fd,SNR1,(1:nn)/fd,SNR2),xlim([23.5 24.0])
figure(4),plot(Y1)
figure(5),plot(Y2)
```



#### Appendix D - Final gunshot detection algorithm

```
#include <Audio.h>
#include <Wire.h>
#include <SPI.h>
#include <SD.h>
#include <SerialFlash.h>
File fftDataFile;
AudioInputUSB
                                usb1;
AudioOutputI2S i2s1;
AudioAnalyzeFFT1024 myFFT;
AudioConnection
                           patchCord1(usb1, 0, i2s1, 0);
patchCord2(usb1, 1, i2s1, 1);
patchCord3(usb1, 0, myFFT, 0);
                                 patchCord1(usb1, 0, i2s1, 0);
AudioConnection
AudioConnection
AudioControlSGTL5000
                               sgt15000_1;
//-----GENERAL VARIABLES-----//
int y=0;
                                            //Frame Counter
                                       //SD-Card Chip
//Pi
const int chipSelect = 10;
double pi = 3.1415926535;
                                           //Seconds Calculation
float Seconds;
float Minutes;
                                           //Minutes Calculation
                                            //Time Counter
unsigned long time;
//-----ENERGY VARIABLES-----//
//-----ENERGY VARIABLES------//
double level[20]; //Current frame energy levels
double lastLevel[20]; //Previous frame energy levels
double diffLevel[20]; //Difference in current levels minus previous
double diffLevelSum; //Sum of FFT Bins for Energy
double diffLevelAvg; //Average of FFT Bins for Energy
double myMax = 0.00000000; //Max Energy Value in Array
double multiplier = 100000.0; //Level Multiplier
int revIndex = 0: //Bin # with Highest Energy
int maxIndex = 0;
                                             //Bin # with Highest Energy
//-----VECTOR OF CHANGE (ENERGY)-----//
double adj = 1024;
                                         //Adjacent/Time variable in Energy Vector Of Change
double opp;
                                         //Opposite/Level variable in Energy Vector Of Change
                    //Hypotenuse/Difference variable in Energy Vector Of
float hyp;
Change
double tangent;
                                       //Tangent Energy (diffLevel / Adj)
                                              //Vector Angle Radians
double thetaRad;
float thetaDeg;
                                             //Vector Angle Degrees
//-----SPECTRAL CENTROID VARIABLES-----//
```



```
double SpectralCentroid;
                              //Spectral Centroid Value
double SpectralCentroidPrev;
                              //Previous Spectral Centroid Value
double diffCentroid;
                              //Different in current SC value minus previous
double SChyp;
                             //Spectral Centroid Hypotenuse
double SCtangent;
                             //Tangent Spectral Centroid
                              //SC Vector Angle Radians
double SCthetaRad;
                              //SC Vector Angle Degrees
double SCthetaDeg;
double den;
                              //Denominator of SC
double num;
                              //Numerator of SC
int myBins[20] = {23, 69, 115, 184, 276, 368, 460, 621, 851, 1196, 1656, 2116,
3496, 5796, 8096, 10396, 12696, 14996, 17296, 21223};
int numba = 1;
//-----//
void setup() {
 //----SETUP TEENSY AUDIO CARD-----//
 SPI.setMOSI(7);
 SPI.setSCK(14);
 Serial.begin(9600);
 AudioMemory(12);
 sgtl5000 1.enable();
 sgt15000 1.volume(0.6);
 //-----SETUP FFT WINDOW TYPE-----//
 myFFT.windowFunction(AudioWindowHanning1024);
 //-----INITIALIZE SD CARD-----//
 Serial.print("Initializing SD card...");
 if (!SD.begin(chipSelect)) {
   Serial.println("initialization failed!");
   return;
 }
 Serial.println("initialization done.");
 //----OPEN SD TEXT FILE-----//
   fftDataFile = SD.open("test.txt", FILE_WRITE);
   Serial.println("Open File Write");
   Serial.println();
}
//-----//
```



```
void loop() {
   //----SET VOLUME PLAYBACK FOR USB AUDIO PLAYBACK FROM
COMPUTER-----//
  float vol = usb1.volume();
 if (vol > 0.5) {
   vol = 0.3 + vol * 0.5;
  }
   //----BEGIN FFT CALCULATION-----//
                            // each time new FFT data is available print it all
   if (myFFT.available()) {
to the Arduino Serial Monitor
   //-----TIMER-----//
   //Serial.print("Milliseconds: ");
   time = millis();
   //Serial.println(time);
   //-----FFT ARRAY-----//
   level[0] = myFFT.read(0);
   level[1] = myFFT.read(1);
   level[2] = myFFT.read(2);
   level[3] = myFFT.read(3, 4);
   level[4] = myFFT.read(5, 6);
   level[5] = myFFT.read(7, 8);
   level[6] = myFFT.read(9, 10);
   level[7] = myFFT.read(11, 15);
   level[8] = myFFT.read(16, 20);
   level[9] = myFFT.read(21, 30);
   level[10] = myFFT.read(31, 40);
   level[11] = myFFT.read(41, 50);
   level[12] = myFFT.read(51, 100);
   level[13] = myFFT.read(101, 150);
   level[14] = myFFT.read(151, 200);
   level[15] = myFFT.read(201, 250);
   level[16] = myFFT.read(251, 300);
   level[17] = myFFT.read(301, 350);
   level[18] = myFFT.read(351, 400);
   level[19] = myFFT.read(401, 511);
     //-----SPECTRAL CENTROID CALCULATION-----//
     for (int i = 0; i < 20; i++){
       num += (level[i]) * myBins[i];
       den += (level[i]);
```



```
SpectralCentroidPrev = SpectralCentroid;
     }
     SpectralCentroid = (num /den);
     if(num > 1){
      num = 0;
      den = 0;
     }
      //----ENERGY CALCULATION-----//
     for (int i = 0; i < 6; i++) {
     diffLevel[i] = ((level[i] - lastLevel[i]) * multiplier);
     diffLevelSum += (diffLevel[i]);
     lastLevel[i] = level[i];
     if(level[i]*multiplier > myMax){
         myMax = level[i]*multiplier;
         maxIndex = i;}
     }
      //-----DIFFERENCE CALCULATION-----//
     diffLevelAvg = (diffLevelSum / 7);
     diffCentroid = (SpectralCentroid - SpectralCentroidPrev);
       //-----VECTOR MAGNITUDE CALCULATION-----//
hyp = (sqrt((pow((diffLevelAvg), 2) + (pow(adj, 2)))); //ENERGY
     //Serial.print("Energy Hyp:");
     //Serial.print(hyp);
     //Serial.println();
SChyp = (sqrt((pow((diffCentroid), 2) + (pow(adj, 2))))); //CENTROID
     //Serial.print("Centroid Hyp: ");
     //Serial.print(SChyp);
     //Serial.println();
       //----- VECTOR DIRECTION CALCULATION-----//
     tangent = (diffLevelAvg / adj);
                                                       //ENERGY
     thetaRad = (atan(tangent));
     thetaDeg = (thetaRad * (180/pi));
     //Serial.print("Energy Angle:");
```



```
//Serial.print(thetaDeg);
      //Serial.println();
      SCtangent = (diffCentroid / adj);
                                                          //CENTROID
      SCthetaRad = (atan(SCtangent));
      SCthetaDeg = (SCthetaRad * (180/pi));
      //Serial.print("Centroid Angle: ");
      //Serial.print(SCthetaDeg);
      //Serial.println();
       //-----Gunshot Detection & Print to Serial Monitor-----//
      if ( (hyp > 1033.0) && (thetaDeg > 8) && (SChyp > 1030.0) &&
(SpectralCentroid < 4500.0) && (SCthetaDeg < -25) )</pre>
     {
       Serial.println("SHOT DETECTED");
       Serial.print("Frame: ");
       Serial.print(y);
       Serial.print(" , ");
       Serial.print("Time: ");
       Seconds = (time / 1000);
       Minutes = (Seconds / 60);
       Serial.print(Minutes);
       Serial.print(" , ");
       Serial.print("Energy Difference: ");
       Serial.print(hyp);
       Serial.print(" , ");
       Serial.print("Centroid Difference: ");
       Serial.print(SChyp);
       Serial.print(" , ");
       Serial.print("Centroid: ");
       Serial.print(SpectralCentroid);
       Serial.print(" , ");
       Serial.print("EnergyAngle: ");
       Serial.print(thetaDeg);
       Serial.print(" , ");
```

Serial.print("Centroid Angle: ");

Serial.print(SCthetaDeg);

Serial.println(); Serial.print(numba); numba = (numba + 1); Serial.println();

}



```
if ( (hyp > 1033.0) && (thetaDeg > 8) && (SChyp > 1030.0) &&
(SpectralCentroid < 4500.0) && (SCthetaDeg < -25) )</pre>
     {
       delay(1000);
      }
      //-----Rolling Average Calculation (NOT IN USE)-----//
      //myRA.addValue(level[0]*multiplier);
      //Serial.println();
     //Serial.print ("RA:");
     //Serial.print(myRA.getAverage(), 3);
      //Serial.println();
     //fftDataFile.print(myRA.getAverage(), 3);
     //fftDataFile.print(",");
     y++;
   }
 }
}
```